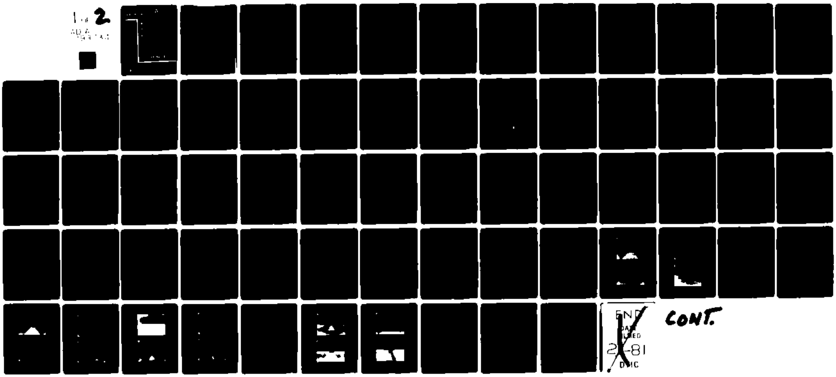
AD-A093 734 APPLIED DIGITAL COMMUNICATIONS MOORESTOWN NJ F/G 5/8
FLIGHT TRAINING SIMULATOR: SURFACE TEXTURING VIA PSEUDO RANDOM --ETC(U)
DEC 80 J F DESPAUTZ, M B BENDER, V M MCNAMARA F33615-77-C-0068

UNCLASSIFIED AFHRL-TR-80-13 NL

1 of 2
AD A
093734



END
DATE
FILMED
2-81
DTIC

CONT.

A
37

AFHRL-TR-80-13

LEVEL II

AIR FORCE HUMAN RESOURCES LABORATORY

AD A093734

FLIGHT TRAINING SIMULATOR: SURFACE TEXTURING VIA PSEUDO RANDOM NOISE CODES

By

Joseph F. deSpautz
Michael B. Bender
Virginia M. McNamara

Applied Digital Communications
214 Flynn Avenue
Moorestown, New Jersey 08057

OPERATIONS TRAINING DIVISION
Williams Air Force Base, Arizona 85224

December 1980

Final Report

Approved for public release; distribution unlimited.

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235

NOTICE

This final report was submitted by the Applied Digital Communications, 214 Flynn Avenue, Moorestown, New Jersey 08057, under Contract F33615-77-C-0068, Project 6114, with the Operations Training Division, Air Force Human Resources Laboratory (AFSC), Williams Air Force Base, Arizona 85224. Michael L. Ingalls was the Contract Monitor for the Laboratory.

This technical report has been reviewed and is approved for publication.

MARTY R. ROCKWAY, Technical Director
Operations Training Division

RONALD W. TERRY, Colonel, USAF
Commander

SUBJECT TO EXPORT CONTROL LAWS

12 65

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFHRL-TR-80-13	2. GOVT ACCESSION NO. AD-A093734	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FLIGHT TRAINING SIMULATOR: SURFACE TEXTURING VIA PSEUDO RANDOM NOISE CODES		5. TYPE OF REPORT & PERIOD COVERED Final rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joseph F. deSpautz, Michael B. Bender, Virginia M. McNamara		8. CONTRACT OR GRANT NUMBER(s) F33615-77-C-0068 ok
9. PERFORMING ORGANIZATION NAME AND ADDRESS Applied Digital Communications, 214 Flynn Avenue, Moorestown, New Jersey 08057		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62205F 61142106
11. CONTROLLING OFFICE NAME AND ADDRESS HQ Air Force Human Resources Laboratory (AFSC), Brooks Air Force Base, Texas 78235		12. REPORT DATE December 1980
		13. NUMBER OF PAGES 64
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Operations Training Division, Air Force Human Resources Laboratory, Williams Air Force Base, Arizona 85224		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

computer image generation, simulation, cyclic codes, texturing, mathematical models, training

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Increased use of infrared and low light level television systems within the operational environment has presented increasingly difficult problems to the technology of simulation systems in the area of target identification, textural cues, and simulation fidelity. The principal effort reported here has been to develop a software simulation using cyclic codes for providing surface texture without the use of typical computer image generation (CIG) edge-type algorithms. The latter type algorithms, when used to provide valid and realistic surface detail, cause the edge processing load to exceed system capability.

Cyclic code techniques provide a compact texture data base definition that is versatile, has substantial tonal assignment capability, and is based on real-time simulation concepts.

DD FORM 1 JAN 73 1473

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

393139

Item 20 (Continued)

The simulator system was delivered and made operational on the Sigma 5 facility using the ACTES facility for offline video data storage, retrieval and display. The software is user interactive from a remote terminal and includes capabilities for modeling the data base, display characteristics, aircraft positioning, and simulator control of printouts and default options.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SUMMARY

PROBLEM

The increased emphasis on the use of simulators for training of infrared (IR) and low light level television (LLLTV) systems has presented some technical simulation problems. One such problem is target identification which requires the need for proper textural cues, especially on low-level, long-range flight profiles. These textural cues aid in providing increased simulator fidelity through enhanced realism and better discrimination between target types.

APPROACH

Computer image generation (CIG) techniques offer solutions to these simulation problems. However, the capacity of typical edge-type algorithms for textural detail rapidly outstrips system capabilities for real-time operations. One alternate technique is the use of cyclic codes for providing surface detail that is spatially correlatable and provides different levels of detail as a function of range. This technique was used on a previous commercial contract to generate a complete seascape in real time on a TV raster-type display. For this AFHRL study, hardware was modeled in software to provide a non-real-time simulation that was applicable to both real-time operations and present CIG-type simulators. CIG-type coordinate transformations, perspective equations, clipping, and edge-smoothing algorithms were written for operation of the simulation on the AFHRL Advanced CIG Techniques Evaluation System (ACTES) facility at Wright-Patterson AFB.

A data base definition schema was developed using the compact notation of cyclic codes, their theoretical properties and algorithms for varying the code as a function of geographic position. The schema is based upon non-edge-type algorithms.

RESULTS

A simulation program was written and made operational on contractor faciliites and the Xerox Sigma 5 and ACTES facilities at WPAFB. The simulation was displayed on the ACTES display using its offline data storage and display capabilities. The simulation demonstrated the types of texture that could be generated from a cyclic code data base. Variation in levels of detail from background to foreground,as well as spatial coherence, was demonstrated

by running various scenarios. Geometric and texturing aspects were demonstrated for various fields of view and aircraft positions with respect to the textured terrain area, showing the applicability of the techniques to both sensor and visual type displays.

CONCLUSIONS

The implementation on the ACTES demonstrates that cyclic coding techniques are feasible from mathematical and software programming viewpoints for simulators typifying operational IR and LLLTV systems. Texture realism was not validated as it was not a part of this study. Future efforts should include three-dimensional aircraft motion, texture realism and validation of the real-time aspects of the implemented software models and algorithms.

PREFACE

The study was initiated by the Advanced Systems Division, Air Force Human Resources Laboratory, Wright-Patterson Air Force Base, Ohio. The effort was conducted by Applied Digital Communications, 214 Flynn Avenue, Moorestown, New Jersey 08057. Mr. Joseph F. deSpautz was the principal investigator for Applied Digital Communications. Captain Michael L. Ingalls was the contract manager for the Air Force Human Resources Laboratory. Mr. Michael Nicol of Air Force Human Resources Laboratory provided the software interface for data recording and display on the ACTES facility.

Applied Digital Communications acknowledges the significant assistance received from Mr. William Schelker and his staff at Air Force Human Resources Laboratory during the software conversion to the Sigma 5 computer and in running the simulation.

1.0 INTRODUCTION

The purpose of this study was to demonstrate the feasibility of using pseudo random noise (PRN) codes for surface texturing in flight training simulators. To accomplish this, a software simulation was developed to produce static and dynamic views of a flat surface from varying viewpoints in simulated flight, illustrating changes in texture detail with changes in perspective. Included among the system requirements were the ability to control amplitude and spacing of tonal variations and to demonstrate flexibility in generating a variety of texture patterns.

Previous applications, using cyclic codes for image enhancement, have used the randomness or incoherent code properties for white noise generation that is added to the scene. This type of texturing is not correlated TV frame to TV frame, and, therefore, is not usable for simulation realism or training cueing. This study approach associates code states to geographic position (or entire scene) and uses the code properties and methods for efficiently generating them, thereby allowing line-to-line and TV frame-to-frame correlation.

The texture is calculated in a deterministic manner and overlays a constant tone ground area. Scene fidelity will be demonstrated by showing the variation in textural detail as a function of sensor altitude and position.

The implementation of PRN coding techniques has already been shown in company programs, including a real-time seascape generator. Cyclic code generators can operate at TV raster generation speeds. PRN structures, tonal variation, refresh rate, etc. can all be controlled in real time with appropriate hardware.

Large area texturing can be readily constructed with a minimum of storage requirements. The initial texture definitions need only be kept in memory for each of the PRN generators, and these values can be eliminated with the construction of a hardware function to calculate the individual starting states in real time.

To demonstrate the texturing capabilities of using PRN techniques for surface texturing, scenarios of level flight over a rectangular terrain area that has a texture were recorded for playback on the AFHRL ACTES facility. To permit a variety of test cases, parameters including terrain size and location, PRN structure. aircraft altitude

and velocity, and field of view are under interactive control of the user. Incorporated in the simulation are the typical operations of spatial-image plane transformations, field-of-view clipping, perspective transformations, pixel intensity calculations, edge smoothing, and display formatting.

The simulation demonstrates several types of textures which are possible with the cyclic code data base. Control of textures is achieved in two ways: (a) the use of repeating patterns that are generated by cyclic codes which are called subterrains permits reinitialization of the basic PRN pattern at various points to introduce desired discontinuities, and (b) the degree of texture detail can be controlled by reducing the range of possible gray levels to any desired number.

Geometric and texturing aspects have been demonstrated from various fields of view of the sensor display and aircraft positions relative to the textured surface. As the range of the aircraft to the data base is decreased, the expected increase in resolvable detail is demonstrated quite accurately. Alterations in display field of view show that the technique is potentially applicable to visual as well as sensor simulations.

2.0 TECHNICAL APPROACH

A number of different techniques have been investigated to provide surface detail or terrain texturing in computer image generation (CIG) for simulation of operational infrared (IR) and low light level television (LLLTV) systems. These techniques include cyclic coding algorithms for tonal generation.

Past company programs have demonstrated that compact cyclic codes could be used to provide a coherent real-time surface texturing of dynamic seascape scenes. This is contrasted to other applications of cyclic code techniques that provide texturing using the random or incoherent code properties (i.e., white noise generator). This latter type of texturing is not spatially correlated and, therefore, is not valid for display realism.

The approach used in this study was

1. To associate code states to geographic position to provide spatial correlation or visual coherence.

2. To develop software models of real-time hardware techniques for the generation of texture for presentation on TV raster-type displays.

3. To develop a data base definition without edges for variable surface texture that is parameter controlled.

4. To develop a non-real-time software simulation for execution in the AFHRL ACTES facility.

5. To demonstrate various texturing concepts by running and recording scenarios on existing AFHRL facilities.

The system flow diagram of the simulation developed is shown in Figure 1 and includes:

1. Data Base Generation.

2. Data Input/Output and Simulation Control.

3. Aircraft Sensor Positioning.

4. Spatial Transformations.

5. Perspective Transformations.

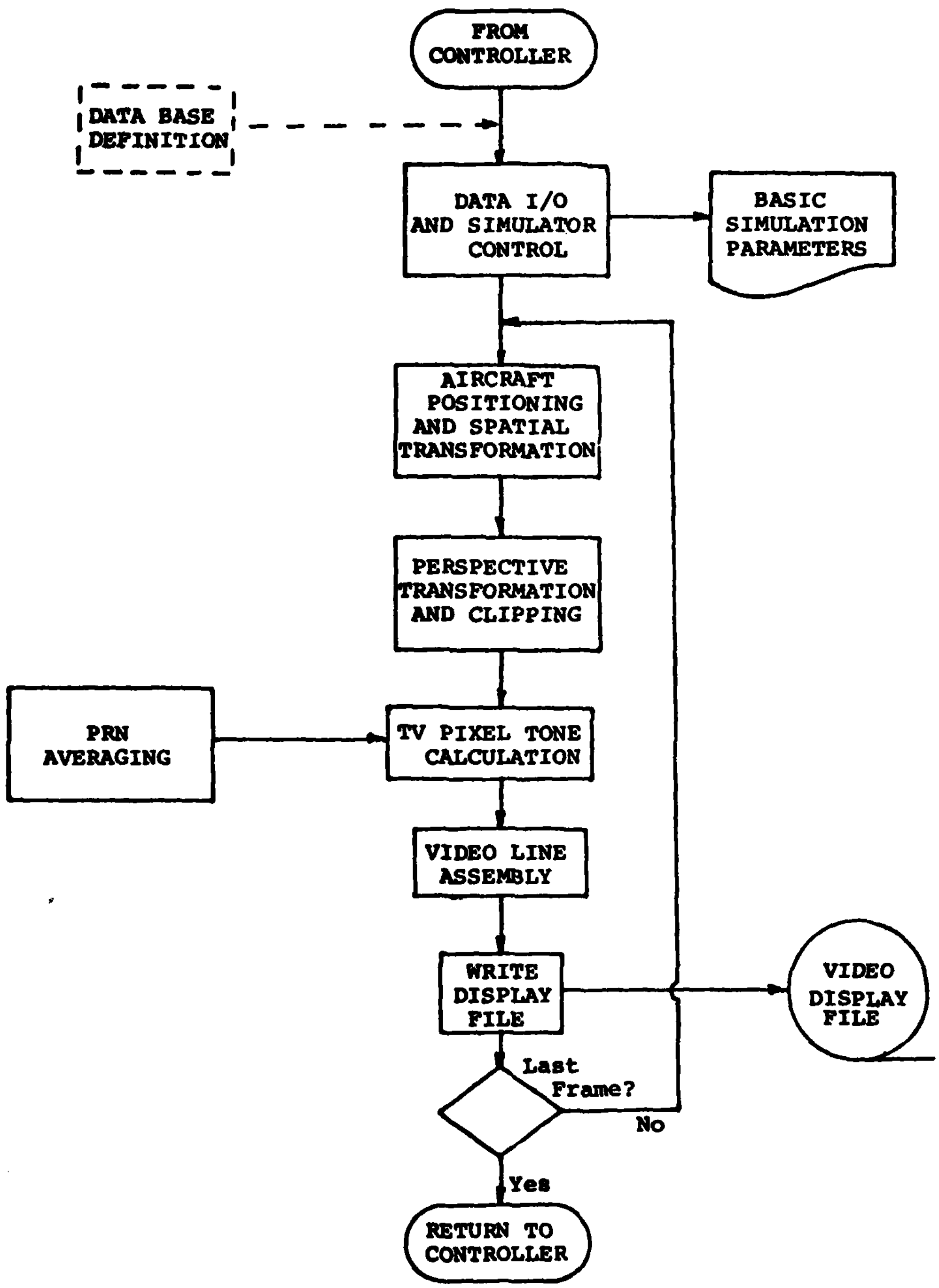


Figure 1. System Flow Diagram

6. TV Raster Pixel Tone Calculation.

7. Video Line Assembly.

Described in this section are the technical approach and mathematical techniques used in the simulation to develop the dynamic TV frame data which was stored on magnetic tape. The data tape uses the offline ACTES facility for visual display.

A discussion of data base construction is presented. Since the terrain texture is defined by using PRN codes to represent texture tones, an analysis of PRN codes and their generation is included. The mapping of these codes onto a terrain surface is described which defines the terrain structure.

The simulation includes the coordinate transformations for independent aircraft motion, clipping of terrain for a variable field of view (FOV) screen with selectable parameters for image display characteristics, and algorithms for determining the visible portion of the terrain within each TV pixel. The initial study was limited to horizontal flybys over the terrain, and rotational effects were investigated under a contract amendment.

An algorithm associates the correct area of the terrain with a given pixel by projecting the pixel definition onto the terrain and determining the location and size of the projection. The resulting composite area tone is the image plane value that is to be displayed if the terrain is not blocked by any other objects.

To provide different levels of texture detail, multiple PRN values fall within a pixel; thus, a technique is provided for averaging these values with a minimum of calculation. For cases in which a terrain edge or boundary falls within a pixel, edge smoothing techniques are applied.

The simulation was optimized in terms of execution time and amount of texturing control. Integer mathematics was implemented to eliminate floating point arithmetic to improve run times. Control of textural tone attenuation was made a program control parameter.

2.1 DATA BASE DEFINITION

2.1.1 CYCLIC CODE CONSIDERATIONS

One of the most important advances in error correcting code development has been concerned with cyclic codes. A cyclic code is one in which each element of a set is obtained by cyclically shifting each predecessor one unit.

Cyclic codes are of considerable importance in CIG enhancements due to their

1. Theoretical properties.
2. Ease of generation.
3. Ease of control.

The cyclic codes chosen for investigation in this study are PRN codes having the following properties:

1. Maximal length
2. Polynomial degree
3. Primitive polynomial characteristics
4. Compact polynomial notation

Maximal length refers to the code set having maximum periodicity; that is, the number of unique states which the code generates before the code is repeated.

Polynomial degree refers to total length of the code sequence and is an indication of the amount of granularity or variation in a sequence.

A primitive polynomial indicates a unique code. A primitive code is not reducible to two other codes.

Polynomials offer the advantage of compact notation. For a maximal length, primitive polynomial of degree m, 2^m-1 code states are generated.

2.1.2 LINEAR AND NON-LINEAR FUNCTIONS

Shift registers with linear feedback functions are the most amenable to analysis. Linear Boolean functions are analogous to linear algebraic functions. Any linear algebraic equation

$$ax + by = c$$

may be solved for y once x is given.

The Boolean equation

$$x + y = 1$$

is a non-linear function of x and y. The + sign here denotes logical addition (OR). Given a state value of

$$x = 1$$

y is indeterminate. Since y cannot be determined for both state values of x, the binary operation of logical addition is referred to as information losing. A linear operation must be information preserving.

Logical multiplication is also non-linear. Given the Boolean equation

$$x \cdot y = 0$$

and the state value

$$x = 0$$

y is indeterminate.

The Boolean operation of addition modulo-2 (exclusive OR) is linear and information preserving. Given the Boolean equation

$$x \oplus y = 1$$

or $\quad x \oplus y = 0$

y may be determined once x is given.

Only linear functions will be considered here.

2.1.3 MATRIX DESCRIPTION OF CYCLIC CODES

Consider the generalized r-stage feedback shift register with a linear feedback function (Figure 2). The stages of the register are designated $S_0, S_1, S_2, \ldots, S_r$. The output of stage S_i is connected to the input of the modulo-2 summer when $C_i = 1$. C_0 is always one, otherwise fewer than r stages would be in use. The external input, e, is a Boolean constant.

Let x_i represent the current state and X_i the next state of stage S_i. The next state X_i of each stage may be expressed as a linear Boolean function of the present state x_i of one or more stages.

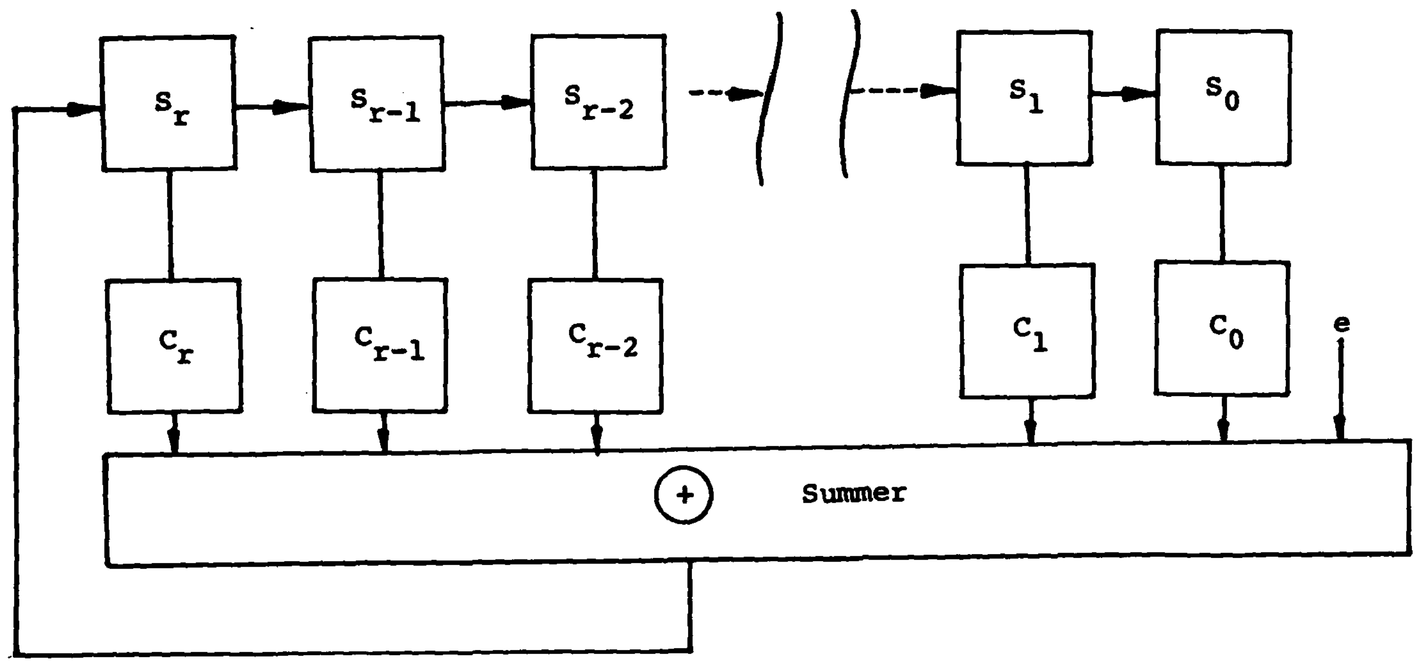


Figure 2. Generalized r-Stage Feedback Shift Register

$$
\begin{aligned}
X_r &= C_r x_r \oplus C_{r-1}\, x_{r-1} \oplus \cdots \oplus C_1 x_1 \oplus C_0 x_0 \oplus e \\
X_{r-1} &= x_r \\
X_{r-2} &= \qquad x_{r-1} \\
&\;\;\vdots \\
X_1 &= \qquad\qquad x_2 \\
X_0 &= \qquad\qquad\qquad x_1
\end{aligned}
$$

This system of equations may be expressed as:

$$
\begin{bmatrix} X_r \\ X_{r-1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ X_2 \\ X_1 \\ X_0 \end{bmatrix}
=
\begin{bmatrix}
C_r & C_{r-1} & \cdots & C_2 & C_1 & C_0 \\
1 & & & & & \\
\cdot & \ddots & & & & \\
\cdot & & \ddots & & & \\
\cdot & & & 1 & & \\
\cdot & & & & 1 & \\
\cdot & & & & 1 & 0
\end{bmatrix}
\begin{bmatrix} x_r \\ x_{r-1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_2 \\ x_1 \\ x_0 \end{bmatrix}
\oplus
\begin{bmatrix} e \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}
$$

or $\quad X = Tx \oplus L$

The r x r Boolean matrix, T, is non-singular since its row (column) vectors are linearly independent. T represents the linear transformation of an r component vector (present state) into another r component vector (next state). L represents a translation. When $e = 1$, the modulo-2 sum of the vector L with Tx represents the complement of the bit fed back.

A linear transformation T followed by a translation L is called an affine transformation. Every translation is one-to-one and has an inverse. The linear transformation

T is one-to-one and has an inverse. Hence, the affine transformation $Tx \oplus L$ is one-to-one and has an inverse. Because of this property, each state has a unique successor.

Of primary interest are those feedback combinations which yield maximal length sequences. Those maximal-length cyclic codes with e = 0 have been investigated.

For the case e = 0,

$$X = T_x$$

An initial state x will be followed by the states

$$T_x,\ T^2x,\ T^3{}_x,\ .\ .\ .\ .,\ T^k{}_x = x,\ T^{k+1}{}_x = T_x,\ .\ .\ .$$

The smallest value of k for which $T^k = I$ is the length of the longest possible cycle. In general, for maximal length sequences

$$k = 2^m - 1$$

where m is the degree of the feedback polynomial.

2.1.4 APPLICATION TO DATA BASE DEFINITION

The data base generation algorithm to create a textured grid steps the initial starting state p_i using the appropriate code generator to yield texture values p_{i+1}, P_{i+2}, etc. The total textured area of the terrain is generated by altering the starting state per grid line and performing the code generation algorithm with the new starting state. An example is to cause the starting state to precess by one state for each line, thereby producing a skewed field. This precession by one code state, yielding a symmetric configuration of codes, has a number of mathematical properties which are presented later in this section. It is the data base texture procedure used in this investigation.

The software which generates these cyclic codes was designed to simulate the actual hardware algorithm. A pseudoshift register was implemented using a table lookup algorithm to decrease software execution time.

2.1.5 TERRAIN STRUCTURE

The size and shape of the terrain that is contained in each pixel projection determines the PRN tones to be combined or averaged to form the pixel tone. There are potentially three types of combinational operations that have to be considered to provide proper pixel tone. These operations are a function of data base quantization and

object-image plane geometry and include the following cases:

1. One PRN code overlays more than one pixel.
2. One PRN occupies one pixel.
3. Many PRN codes and portions thereof occupy a pixel.

The latter case is the principal case investigated in this study as it shows the texture variation as a function of range and is representative of realistic conditions.

Figure 3 shows the representative geometry of the terrain structure. The data base is a rectangular area identified by its vertices in the X,Y plane. The vertices form edges for use in the perspective and clipping algorithms. The data base is quantized into a grid in which each grid element defines a "minimum texture area" (MTA). Each of these MTA elements is associated with a PRN value representing the texture tone. The terrain data base lying on the surrounding ground has a higher priority than the surrounding ground and, therefore, blocks it in the image plane. In the simulator, the data base size and MTA size are simulation variables controlled by input parameters.

Each PRN value is associated with a tonal intensity for each MTA. A row of MTAs parallel to the Y-axis forms a terrain line. The assignment of PRN values to a terrain line is made by specifying a "starting state" for the first MTA on that line. A starting state is an index into the sequence of PRN values; it is used to associate a particular PRN value with the MTA to which it is assigned.

Within a group of adjacent terrain lines called a "subterrain," starting states are increased by one from line to line. This arrangement permits the use of techniques for estimating the average tone of multiple MTAs. By varying the size of subterrains, as well as the starting state differences between adjacent subterrains, the overall texture of the entire terrain can be altered.

The subterrain definition is illustrated in Figure 3. The first subterrain has a starting state of 1 assigned to the first terrain line. Within it, starting states differ by a slip rate of one. The final starting state of the first subterrain is five. For the next subterrain, the initial starting state was slipped 5 states. Within the subterrain, however, the state of each terrain line again is incremented by one state continuing the same pattern as

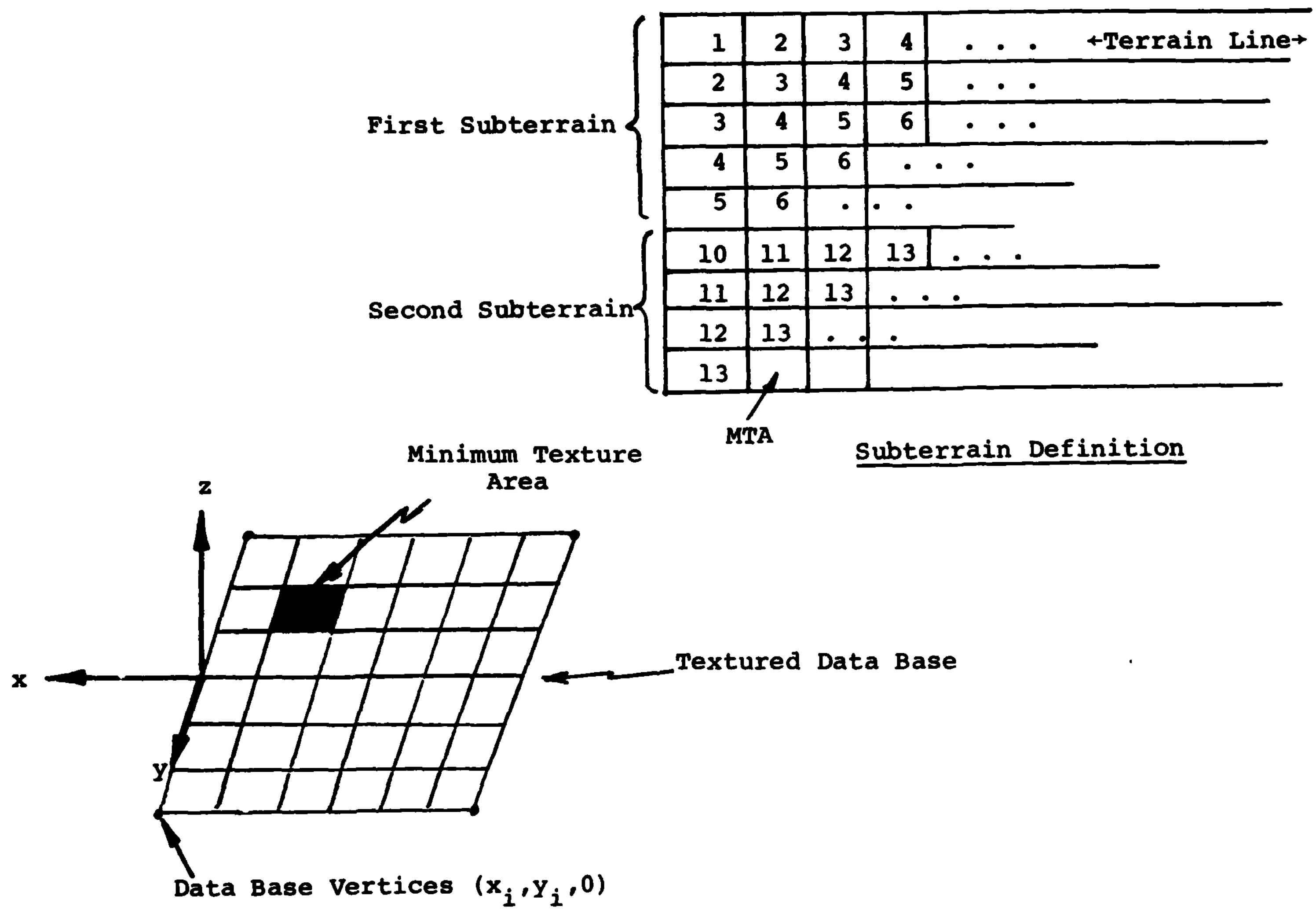


Figure 3. Data Base Definition

in the first subterrain but with different codes.

Using this feature, different patterns could be generated from the same PRN code and simulation algorithms, thereby showing the capabilities of the texturing concepts.

The actual size and shape of the texture area for the straight-and-level flight scenarios that were demonstrated in this study are long and narrow quadrilaterals. Because the projections of the pixels that form a TV line are parallel to the fixed coordinate system, the actual texture area is approximated by rectangles whose coordinates are determined from the perspective equations.

The approximation was believed valid for two reasons. First, because the study was to investigate the case where many PRNs occupy a single pixel, the averaging techniques would account for the partial portions of MTA values without causing scintillation. Second, for the fields of view under investigation, the pixel texture areas were very long narrow figures closely approximating rectangles except at the boundaries. The boundaries were edge smoothed to account for these non-linearities.

2.1.6 AVERAGE TONAL VALUE ESTIMATION

For the case where many PRN values were averaged to form a pixel tone, an algorithm was developed to provide a combinatorial process that could be implementable in suitable hardware for real-time operations. This was accomplished using linear estimation theory. Higher order estimators were not considered as their hardware counterpart may not be suitable for real-time operations.

Let $S_{ij}(n)$ denote a "tonal group" or matrix of n x n MTA elements, at a particular location (i,j) in the PRN data base. Base element p_{ij} is defined as the element in the upper left corner of $S_{ij}(n)$. The average tone of $S_{ij}(n)$ is denoted by $A_{ij}(n)$.

Since direct evaluation of $A_{ij}(n)$ may be time consuming for large values of n, the approach here is to seek approximations to $A_{ij}(n)$ of the form

$$A_{ij}(n) \approx a\, p_i + b$$

where a and b are the correlation coefficients operating on a single element p_i of the $S_{ij}(n)$ tonal group. The

element p_i denotes a unique element of the tonal group set $\{p_{ij}\}$. These linear estimators are derived in the sense of classical least squares estimators. It will be demonstrated that these estimators do exist and have several remarkable properties within the set of all maximal length polynomials of degree m.

Since $S_{ij}(n)$ is symmetric, there exist only $K = 2^n - 1$ unique average tonal values for each choice of n. This reduced set of $A_{ij}(n)$ will be denoted by

$$A_i(n)$$

The classical least squares estimator will be derived for $n = 2$.

Since $n = 2$,

$$A_i(2) = \frac{p_i + 2p_{i+1} + p_{i+2}}{4}$$

The difference between the actual average tonal value and the estimated tonal value is defined as

$$\varepsilon_{ik} = A_i(2) - (a_k\, p_{i+k} + b_k)$$

The sum of the squares of the differences is expressed as

$$\delta_k = \sum_i \varepsilon^2{}_{ik} = \sum_i [A_i(2) - a_k\, p_{i+k} - b_k]^2$$

The more general form p_{i+k} is included here since the correct element of the code set $\{p_i\}$ for estimation may not be the base element of $S_{ij}(n)$. The complexity introduced by the form p_{i+k} now leads to performing k regressions instead of just one regression. Also the value of k must be determined in addition to the constants a_k and b_k.

The conditions that δ_k be a minimum are

$$\frac{\partial\, \delta_k}{\partial\, b_k} = \sum_i A_i - a_k \sum_i p_{i+k} - n\, b_k = 0$$

and

$$\frac{\partial\, \delta_k}{\partial\, a_k} = \sum_i A_i\, p_{i+k} - a_k \sum_i p_{i+k} - b_k \sum_i p_{i+k} = 0$$

This system of equations may be written as

$$\begin{bmatrix} \sum_i p_{i+k} & n \\ \sum_i p^2_{i+k} & \sum_i p_{i+k} \end{bmatrix} \begin{bmatrix} b_k \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_i A_i \\ \sum_i p_{i+k} \cdot A_i \end{bmatrix}$$

Let C be the coefficient matrix of $\begin{bmatrix} b_k \\ a_k \end{bmatrix}$

The solution set is obtained using Cramer's Rule:

$$a_k = (\sum_i A_i \cdot \sum_i p_{i+k} - n \sum_i p_{i+k} \cdot A_i)/\det C$$

$$b_k = (\sum_i p_{i+k} \cdot \sum_i p_{i+k} \cdot A_i - \sum_i p^2_{i+k} \cdot \sum_i A_i)/\det C$$

where $\det C = (\sum_i p_{i+k})^2 - n \sum p^2_{i+k}$

For all cyclic, maximal length polynomials of degree m, det C is invariant since the summations include all members of the code set $\{p_i\}$.

Since the estimator for $A_i(1)$ is

$$A_i(1) = a \cdot p_i + b$$

where $a = 1$

$b = 0$

the correct estimator for $A_i(n)$, $n > 1$ will be found from the set $[a_k, b_k]$ for that value of k for which a_k is closest to 1 and b_k is closest to zero.

An examination of all values of a_k, b_k for all eighth-degree polynomials indicates that the correct value of k is that which causes p_{i+k} to be the element of the secondary diagonal of $S_{ij}(n)$. This is consistent with intuitive expectation since the contribution of the secondary diagonal is weighted most heavily.

There are eight maximal length polynomials of degree 8. For each polynomial a search for a best estimator yielded a and b coefficients that were identical for each tonal order n of $S_{ij}(n)$. This means that the estimation procedure developed for this simulation is independent of the PRN code or textural definition of the data base. Other codes providing different textural detail could be run using the same simulation software.

2.2 DATA I/O AND SIMULATION CONTROL

The simulator was designed to accept input data from several input media: cards, disk or an interactive display terminal. The easiest method (and most popular) was found to be input via the CRT terminal using a video interactive menu with system prompts. The program operator need only respond to questions displayed on the CRT terminal.

The data input routine enables the operator to have complete control over the simulation. The data were divided into four main areas: data base parameters, simulation parameters, video display parameters, and aircraft parameters.

The significant parameters which are under user control include:

- DATA BASE

  Polynomial Generator - PRN size, and configuration

  Shading - Intensities of sky and ground
Number of intensities within data base
Center or average intensity of data base

  Terrain Description - Terrain Size
Terrain Resolution
Subterrain Structure

- DISPLAY

  Size of sensor display screen
Raster display resolution

- SIMULATOR

  Time Scaling - Refresh rate of screen
Length of flight

  Scale Factors - Integer mathematical routines
Internal Smoothing

Printout Controls - Control over which variables are to be printed

Rate at which reports are generated

- AIRCRAFT

  Position, Velocity, Attitude (roll, pitch, yaw)

These variables provide the user with a method for describing the total working environment of the software and simulation scenario.

The simulator outputs to two primary units; video data are put onto nine-track magnetic tape, and the reports are stored on an online disk. The tape data are in a format compatible with the video line assembler of the ACTES facility. Reports are kept on disk primarily to increase simulator efficiency and to avoid tie-ups on the line printer.

2.3 SPATIAL TRANSFORMATIONS

As illustrated in Figure 4, the display window coordinate system is defined by the right-hand coordinate system, u, v, and w. The u axis is perpendicular to and intersects the display plane at its center. The v axis is in the direction of scanning a display line, and the w axis is in the direction of successive scan lines for a TV raster type display. The origin of the u, v, w coordinate system represents the location of the aircraft sensor display focus or viewpoint.

The fixed world coordinate system is designated by the right-hand coordinate axes x,y,z. It is necessary in the simulation to represent a point (x,y,z) in the fixed world system from the viewpoint of the aircraft; that is, to determine the u, v, w coordinates of the point (x,y,z). To accomplish this, the origin of the x,y,z system, (x_o, y_o, z_o), is first translated into $(x_o', y_o', z_o'$, which is the displacement of (x_o, y_o, z_o) in respective axis directions to the location of the aircraft sensor display focus. Then, for any point (x,y,z) a point (x',y',z') is found relative to (x_o', y_o', z_o'):

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_o' \\ y_o' \\ z_o' \end{bmatrix}$$

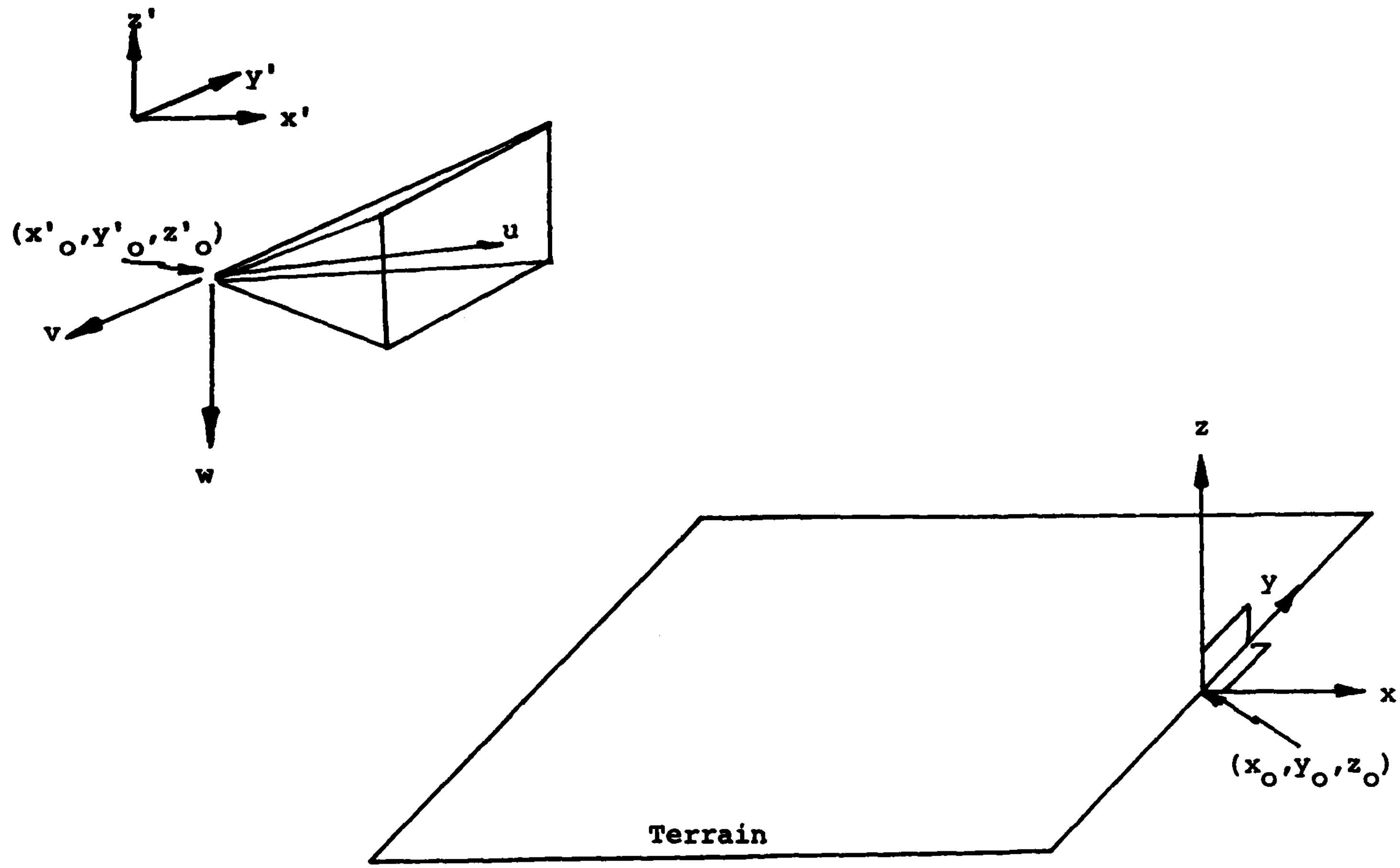


Figure 4. Display Window and Fixed World Coordinate Systems

Lastly, to account for rotation of the aircraft coordinate system relative to the fixed-world system, the matrix

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

is pre-multiplied by a coordinate transformation matrix:

$$A = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{bmatrix}$$

Thus,

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

where the components of A are the direction cosine relationships with respect to three independent angles: heading, pitch, or flightpath angle, and roll.

The point u,v,w is the three-dimensional image space value which is to be clipped, if necessary, and mapped into the two-dimensional sensor image plane.

Point (x_o', y_o', z_o') represents the location of the viewpoint with respect to the origin (x_o, y_o, z_o) of the fixed-world system. Aircraft position will vary from TV frame to TV frame; thus, for each frame, a new position (x_o'', y_o'', z_o'') is calculated from the previous position (x_o', y_o', z_o') as follows:

$$x_o'' = x_o' + v_x \, dt$$

$$y_o'' = y_o' + v_y \, dt$$

$$z_o'' = z_o' + v_z \, dt$$

where v_x, v_y, v_z are aircraft velocity vectors in the x,y and z directions, and dt is the time interval between TV frame calculations.

2.4 CLIPPING

In the clipping algorithm, the terrain is represented by an ordered sequence of points; consecutive pairs of these points comprise terrain edges. Each time the terrain is clipped against a particular clipping plane, a new sequence of points results.

The mathematics of computing a point of intersection, P_i, between a clipping plane and a terrain edge (represented in the three-dimensional coordinate system of the aircraft) is as follows:

$P_i = (u_i, v_i, w_i)$
$P_1 = (u_1, v_1, w_1)$ = first endpoint of terrain edge
$P_2 = (u_2, v_2, w_2)$ = second endpoint of terrain edge

$|P_1P_i|$ = distance from P_1 to P_i

$|P_1P_2|$ = distance from P_1 to P_2

$$u_i = u_1 + r(u_2 - u_1)$$

$$v_i = v_1 + r(v_2 - v_1)$$

$$w_i = w_1 + r(w_2 - w_1)$$

where $0 \leq r \leq 1$, and $r = \dfrac{|P_1P_i|}{|P_1P_2|} = \dfrac{|P_1'P_i'|}{|P_1'P_2'|}$

P_1' and P_2' are illustrated in Figure 5a and defined below:

$P_1' = (u_1', v_1')$ = projection of P_1 onto u-v plane

$P_2' = (u_2', v_2')$ = projection of P_2 onto u-v plane

For the right clipping plane illustrated in Figure 5b, r is computed using basic geometric relationships to be the following:

$$r = \frac{|v_1' - u_1' \tan\beta|}{|v_1' - u_1' \tan\beta| + |v_2' - u_2' \tan\beta|}$$

where β = 1/2 horizontal FOV.

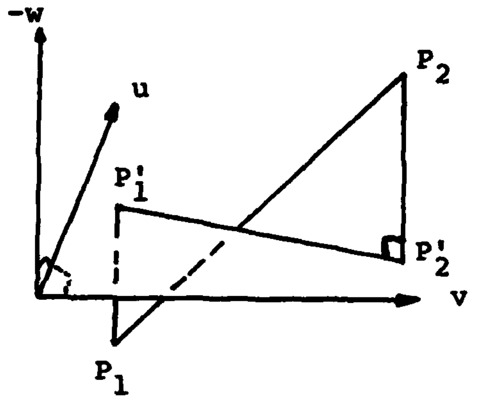


Figure 5a. Projection of P_1, P_2 onto u-v Plane

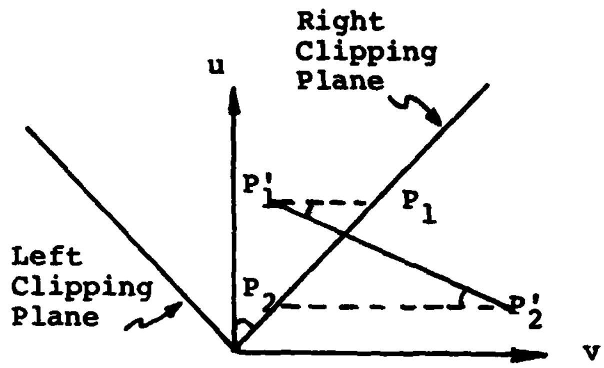


Figure 5b. View onto u-v Plane

Similarly, for the left clipping plane r can be determined from:

$$r = \frac{|v_1' + u_1' \tan \beta|}{|v_1' + u_1' \tan \beta| + |v_2' + u_2' \tan \beta|}$$

For top and bottom clipping planes, respectively,

$$r = \frac{|w_1' + u_1' \tan \alpha|}{|w_1' + u_1' \tan \alpha| + w_2' + u_2' \tan \alpha|} \qquad \text{(top plane)}$$

$$r = \frac{|w_1' - u_1' \tan \alpha|}{|w_1' - u_1' \tan \alpha| + |w_2' - u_2' \tan \alpha|} \qquad \text{(bottom plane)}$$

where

α = 1/2 vertical FOV,

$P_1' = (u_1', w_1')$ = projection of P_1 onto u-w plane,

and $P_2' = (u_2', w_2')$ = projection of P_2 onto u-w plane

As a result of clipping, the vertices which define the terrain in the three-dimensional viewpoint coordinate system all lie within the field of view. Thus, they can be projected onto the plane of the screen without exceeding the limits of the screen itself.

Scan lines are numbered from -239 at the top of the screen to +240 at the bottom, and correspond to the I coordinate of the screen points (see Figure 6). Horizontal elements are numbered from -320 at the left of the screen to +320 at the right and correspond to the J coordinate of the screen points.

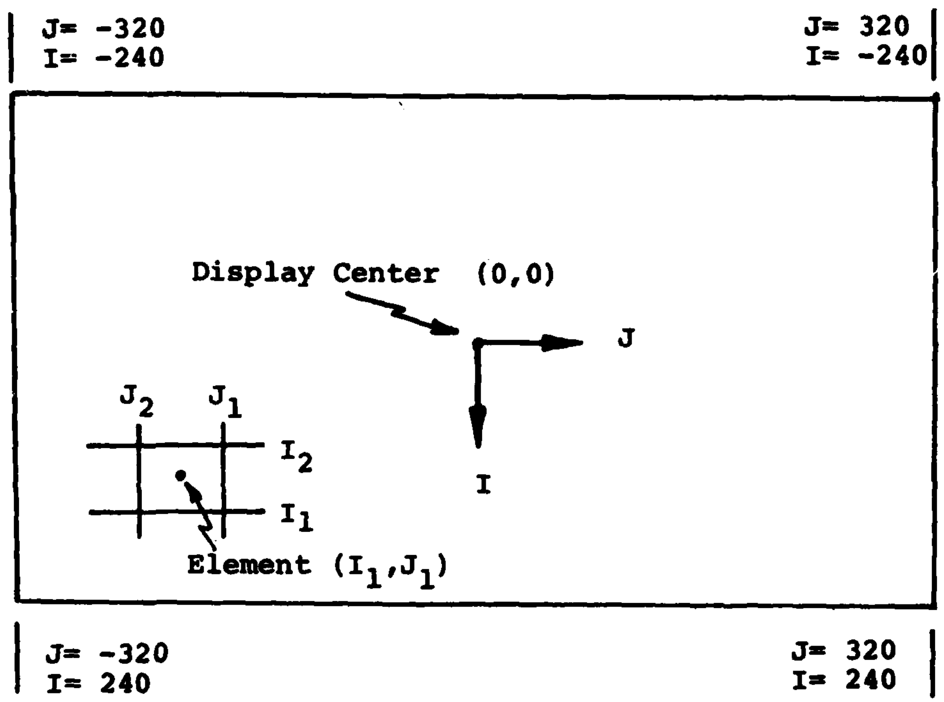


Figure 6. Display Plane Viewed from Focus or Observer's Position

The equations which project a point (u',v',w') onto the screen are as follows:

$$I = d \cdot \frac{w}{u} \qquad J = d \cdot \frac{v}{u}$$

$$\text{where } d = 320 \cdot \cot\left(\frac{\text{horizontal field of view}}{2}\right)$$

$$= \text{distance from viewer to screen in screen units}$$

After determining the screen coordinates of terrain vertices, the slope and J-intercept of each screen edge can be found. This information is needed in determining where edges are crossed by scan lines.

$$\text{Slope} = \frac{J_2 - J_1}{I_2 - I_1}$$

$$\text{Intercept} = J_2 - (\text{Slope}) \cdot I_2$$

where (I_1,J_1) and (I_2,J_2) are endpoints of an edge.

2.5 PIXEL TONE CALCULATION

The primary purpose of this function is to average the appropriate MTA-PRN tones which map into a single picture element or pixel.

Tonal calculation is performed when part (or all) of a TV line falls within the terrain data base; it is performed on a line-by-line basis. The variables which the function requires for its calculations are set up. These include the screen and ground coordinates of the first pixel to fall within the terrain boundaries, the location and size of the terrain area contained in the pixel, the screen location of the last pixel to be averaged, and edge smoothing constants.

Once these variables have been constructed, an algorithm averages the various PRN tones which fall into one pixel. After a tone for an individual pixel has been computed, it is scaled and stored in the TV data line in its appropriate location. The process is repeated for each pixel on the TV line.

When one TV line has been computed, edge smoothing of the terrain is performed and the final tones stored for printing if the print control parameters are satisfied.

2.5.1 PIXEL-TERRAIN GEOMETRY

Figure 7 shows the coordinate systems used, the fixed-world coordinate system (x,y,z) which defines the terrain location, and the viewpoint coordinate system (u,v,w) which defines the viewpoint locations. As shown in Figure 7, the (u,v,w) system is a translation where u is parallel to x, v is parallel to y, and w is parallel to z. The rotated screen coordinate system specified as (u',v',w')

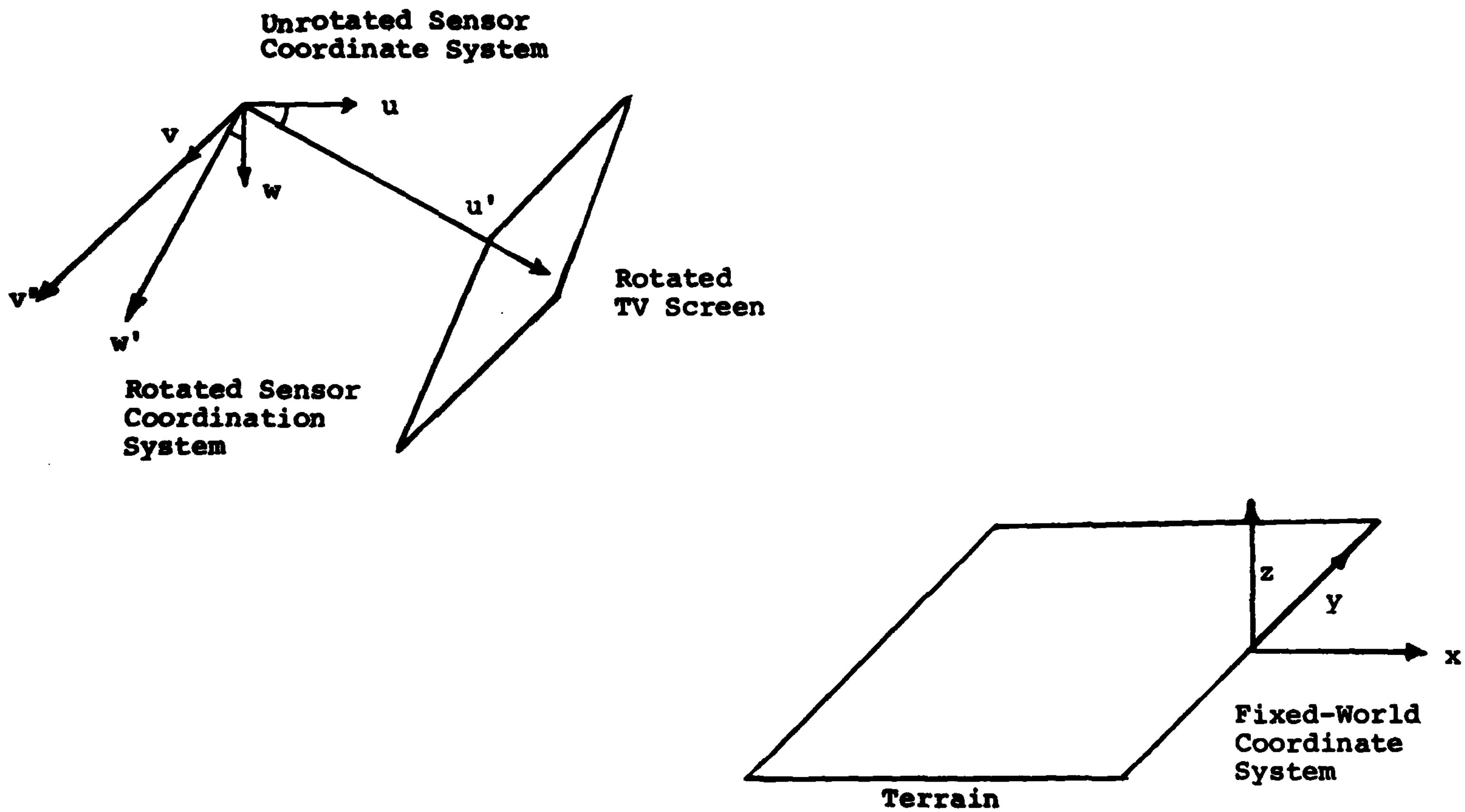


Figure 7. Simulator Coordinate System Definitions

where the u' axis is perpendicular to, and passes through, the geometric center of the screen. v' is parallel to the scan lines, w' is parallel to the screen vertical axis.

Transformation between the rotated screen coordinate system and the unrotated sensor coordinate systems is accomplished using the equation:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \\ A \\ \\ \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

where the components of [A] are the directional cosine relationships with respect to the three independent angles of heading, pitch, and roll.

Mapping of pixel locations to terrain coordinates is accomplished via vectorially projecting image plane coordinates onto the flat earth using the equations

$$x_p = z_p \cdot w_p/u_p$$

$$y_p = z_p \cdot v_p/w_p$$

where z_p is the aircraft altitude, and u_p, v_p and w_p are the coordinates of a corner of a pth pixel in the (u,v,w) system.

2.5.2 TONE COMPUTATION

Tonal groups have been developed which consist of N x N MTA elements. The average value of the tonal group is estimated using a linear estimator, which reduced computation time considerably at a minimum loss of accuracy.

To accommodate image plane rotation, tonal groups will be shifted to achieve the optimum MTA to pixel mapping. Figure 8 illustrates the two basic tonal group geometries. Figure 8a shows tonal groups which are stacked vertically and permitted to shift horizontally ("Free-Y" tonal groups). Figure 8b shows groups which are stacked horizontally and shifted vertically ("Free-X" tonal groups). This description of the algorithm will examine Free-Y tonal group development. Free-X groups developed utilize the same algorithm.

The size and number of tonal groups are determined using the relationships:

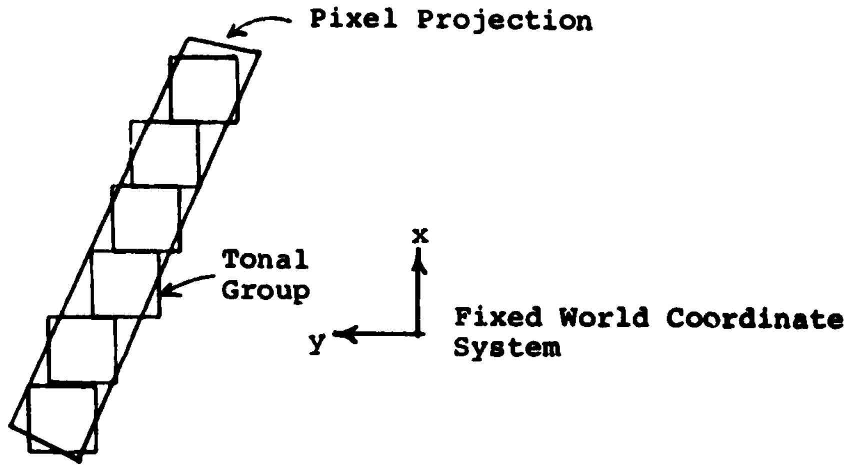


Figure 8a

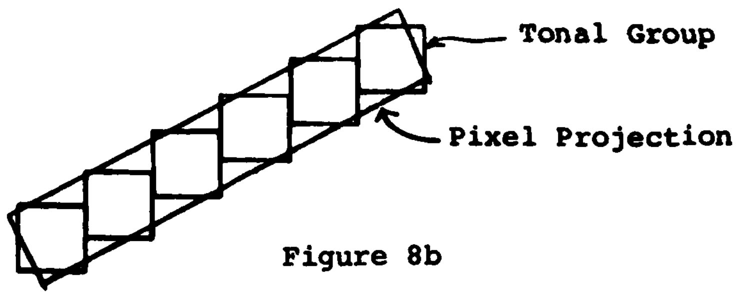


Figure 8b

8a. Tonal groups which are shifted horizontally (Free-Y)

8b. Tonal groups which are shifted vertically (Free-X)

Figure 8. Tonal Group Mapping

$$\frac{\Delta Y_{TG}}{A_{TG}} = \frac{\Delta Y_{PIX}}{A_{PIX}}$$

and

$$\frac{\Delta X_{TG}}{A_{TG}} = \frac{\Delta Y_{PIX}}{A_{PIX}}$$

where

ΔY_{TG}; ΔX_{TG}	are the linear distances covered by the tonal groups which will map into the pixel projection.
ΔY_{PIX}, ΔX_{PIX}	are the linear distances covered by the pixel projections.
A_{TG}	is the area of the tonal groups.
A_{PIX}	is the area of the pixel.

Pixel projections consist of M tonal groups comprised of N x N MTA elements; therefore,

$$\Delta Y_{TG} = N \cdot LMTA$$

$$\Delta X_{TG} = M \cdot N \cdot LMTA$$

$$A_{TG} = M \cdot N^2 \cdot LMTA^2$$

where LMTA is the length of one side of an MTA element.

From these relationships, the size (N) and number (M) tonal groups are found to be

$$N = A_{PIX} / (LMTA \cdot \Delta X_{PIX})$$

$$M = A_{PIX} / (LMTA \cdot \Delta Y_{PIX} * N)$$

Vertical placement of tonal groups is determined as follows. The total height of the combined tonal groups is computed. This value is subtracted from ΔX. This value represents the difference in the vertical distance covered by the pixel projection and the tonal groups. This difference is divided evenly between the top and bottom of the pixel projection, yielding the vertical position of the tonal group stack. This fixes the vertical (X) position of every tonal group as each must lie directly on top of the one below it.

Each individual tonal group is then permitted to float in the Y direction (thus the name "Free-Y") to best align it within the pixel projection. The Y coordinate is computed by substituting the X coordinate of the center of the tonal group into the equation of the pixel centerline.

2.5.3 EDGE SMOOTHING

Edge smoothing is called on a line-by-line basis to compute tones of pixels which contain both terrain and ground. All computations are done using image plane coordinates rather than the fixed-world (terrain) coordinates. The equations of the terrain edges on the screen are set up for each frame prior to any tone computations and stored for their use here.

The tones of these pixels are computed using the tone x area relationship:

$$\text{Tone} = Y_{GND} \cdot \text{IGND} + Y_{TER} \cdot \text{TER}$$

where

Tone	=	new, smoothed tone for a pixel
Y_{GND}	=	fraction of pixel that contains ground
Y_{TER}	=	fraction of pixel that contains terrain
IGND	=	tone of the ground
TER	=	tone of the terrain in that pixel

The fractions Y_{GND} and Y_{TER} are computed using geometric and trigometric relationships of a line (the terrain edge) cutting a square (the pixel). Floating point math is employed for maximum accuracy.

2.6 ADDITIONAL CONSIDERATIONS

Additional functions were established to enhance simulator operation. They are the use of integer mathematics for speed enhancements and gray scale attenuation to provide control of the texture contrast levels. The following paragraphs describe these functions.

An analysis of the simulator timing model showed that floating point instructions in various locations of the software were increasing the run time of the simulator considerably. Integer mathematics were then instituted as follows.

The floating point numbers were pre-scaled using a scale factor (SFACT) in which:

$$\text{SFACT (Sacle Factor)} \equiv 2^{\text{RASSUB}} = \text{SHIFT (1, RASSUB)}$$

where

RASSUB is an input parameter, and

SHIFT is a function by which the first argument is shifted to the left a number of bits equal to the second argument.

More precisely, $A = \text{SHIFT } (M,N) - M \cdot 2^N$ or

$$\text{SFACT} = \text{SHIFT (1,RASSUB)} = 1 \cdot 2^{\text{RASSUB}}$$

After scaling, mathematics is performed in an integer mode, then numbers are restored to their original dimension. Restoring is accomplished (in the general case) by:

$$A = \text{SHIFT } (M,-N) = M \cdot 2^{-N}$$

A negative value for the second argument indicates a shift to the right.

Gray scale reduction is an enhancement developed to provide user control to attenuate the texture contrast viewed in any given scenario. This control is also needed to permit various polynomial degrees to be used without major changes to the software. The ACTES display is capable of 256 gray scales, and an eighth-degree polynomial has 255 individual tones so that, if desired, all tones could be produced on the display.

The algorithm associates actual tones into groups, where the number of groups is an input parameter. The tones are linearly scaled into these groups by simple division. Since this process had to be accomplished for every pixel tone, binary division (shifting) was implemented on the algorithm. The tones were shifted, leaving enough high order bits to associate them with the correct tonal group. This allows the full range of PRN tones to be attenuated about a central value in increments of 2, 4, 8, 16, etc. Other PRN generations having more or fewer tones than the display can be handled by this algorithm.

2.7 VIDEO LINE ASSEMBLY

Data for a TV frame display are calculated on a line-by-line basis. For TV lines which correspond to sky or background only, two buffers containing the appropriate tones are initialized at the start of the simulation. For each TV line which crosses the terrain area, pixel intensity calculations are performed and overlay the 640-element buffer preset to the ground tone.

Every line buffer is prefixed by a frame number and line number, and is passed to a software video interface provided by AFHRL personnel. The interface routine converts data words received from the Sigma 5 to a format acceptable to the ACTES facility for offline video display generation and buffer storage.

3.0 RESULTS

The ability to produce moving pictures with realistic textural cues is an important requirement of training programs and target discrimination applications. The objective of this study was to investigate a texture-producing technique using PRN code states to represent texture tones over a flat area of terrain. A software model was constructed to simulate hardware algorithms for computing pixel tones.

The software simulation has been demonstrated to be an effective tool in generating a variety of controllable textures for CIG, which are spatially correlated to insure realistic displays. Moreover, the operating simulation provides flexibility and interactive control to a user to explore and evaluate the effects of modifying the texture data base definition and gray scale attenuation of a displayed texture.

Using the software simulator, scenarios were produced to simulate views from a sensor moving over surfaces of varying textures. It was shown that as the sensor moves closer to the surface, increased texture detail and contrast are observed. Dynamic imagery demonstrated smooth movement of the viewpoint over the terrain.

The significance of the simulation is that a technique for producing spatially correlatable textures has been validated. Moreover, the algorithms developed can be implemented in hardware for real-time applications.

3.1 EVALUATION OF PRN CODE TECHNIQUES

The principal effort of the study was to develop and investigate the combinatorial algorithms in which large numbers of MTA-PRN tones are averaged to form a pixel intensity. An alternate means to summing and averaging each individual element was analyzed and subsequently implemented in software. A stepped estimator was selected for investigation as it represented a simple algorithm with minimum computation. A stepped estimator is a linear estimator in which the regression coefficients are fixed (i.e., A=1,B=0).

The basis for selecting a candidate technique was that it should be amenable to hardware suitable for real-time operations. Linear and stepped estimators are of this form. The linear estimator has the form

$$T_i = A\, P_{i+k} + B$$

where $T_{i(i=1-255)}$ represents all the combinations within the sequence

P_{i+k} represents a PRN tone within the sequence where k is the displacement from the initial starting state

A,B are regression coefficients of the best fit estimator

A regression analysis was used over the complete code sequence to determine the values of A and B for each value of k, the displacement from the initial starting state. The resulting estimator for the sets of values (k,A,B) represents the best linear estimator. It enables an N x N texture area to be approximated by a linear combination of a single PRN value. Statistics for the estimator were generated to compare the approximation tone to the actual tone to determine the best values of A,B, and k. Statistics were also generated for the stepped estimator.

Table 1 shows the results of the linear regression analysis for a 2 x 2 texture area for one of the eighth-degree primitive polynomials. The k value is given in column DSP. The values in column LEVAR are a measure of the least square error between approximating tone and actual tone for all 2 x 2 sequences using the kth value. A and B are the repression coefficients for the kth value. The column LEVAR shows the incoherent properties of PRN codes. Beyond the first few values of k, the error reaches a maximum and remains constant.

Similar regressions were run for other N x N texture values to develop the values for k, A, and B. It was shown that the off-diagonal term, k = N - 1, for all areas of the form N x N had minimum error. These values were programmed in the simulator.

All eighth-degree maximal length polynomials were tested and the values of k, A, and B for the best estimator were identical. Thus, a single set of linear coefficients could be applied to all maximal-length codes of a given degree. This implies that the hardware algorithm for a specific degree PRN polynomial can be used for all maximal length polynomials of the same degree.

The linear estimator yields a hardware implementation of the form shown in Figure 9. The coefficient K_A represents

TABLE 1. SAMPLE COEFFICIENT SEARCH

POLYNOMIAL IS: 100101011
SIZE = 2X2

DSP	LEVAR	A-COEFF	B-COEFF
0	471982.1562	0.557309	56.664421
1	130035.9219	0.747039	32.379051
2	471982.2500	0.557309	56.664421
3	798392.4375	0.272697	93.094780
4	877670.5625	0.130339	111.316628
5	896325.4375	0.059056	120.440872
6	900399.5000	0.023206	125.029686
7	901111.3750	0.004863	127.377487
8	901107.9375	-0.005141	128.658081
9	900996.7500	-0.010329	129.322098
10	900951.3750	-0.011811	129.511810
11	900950.7500	-0.011811	129.511810
12	900951.4375	-0.011811	129.511810
13	900950.7500	-0.011811	129.511810
14	900951.2500	-0.011811	129.511810
15	900951.2500	-0.011811	129.511810
16	900951.0625	-0.011811	129.511810
17	900951.5000	-0.011811	129.511810
18	900951.3125	-0.011811	129.511810
19	900951.4375	-0.011811	129.511810
20	900950.9375	-0.011811	129.511810
21	900951.1875	-0.011811	129.511810
22	900950.8750	-0.011811	129.511810
23	900951.2500	-0.011811	129.511810
24	900951.1875	-0.011811	129.511810
25	900951.3750	-0.011811	129.511810
26	900951.6250	-0.011811	129.511810
27	900951.6250	-0.011811	129.511810
28	900951.5625	-0.011811	129.511810
29	900951.2500	-0.011811	129.511810
30	900951.3750	-0.011811	129.511810
31	900951.5625	-0.011811	129.511810
32	900951.1250	-0.011811	129.511810
33	900951.8750	-0.011811	129.511810
34	900951.3125	-0.011811	129.511810

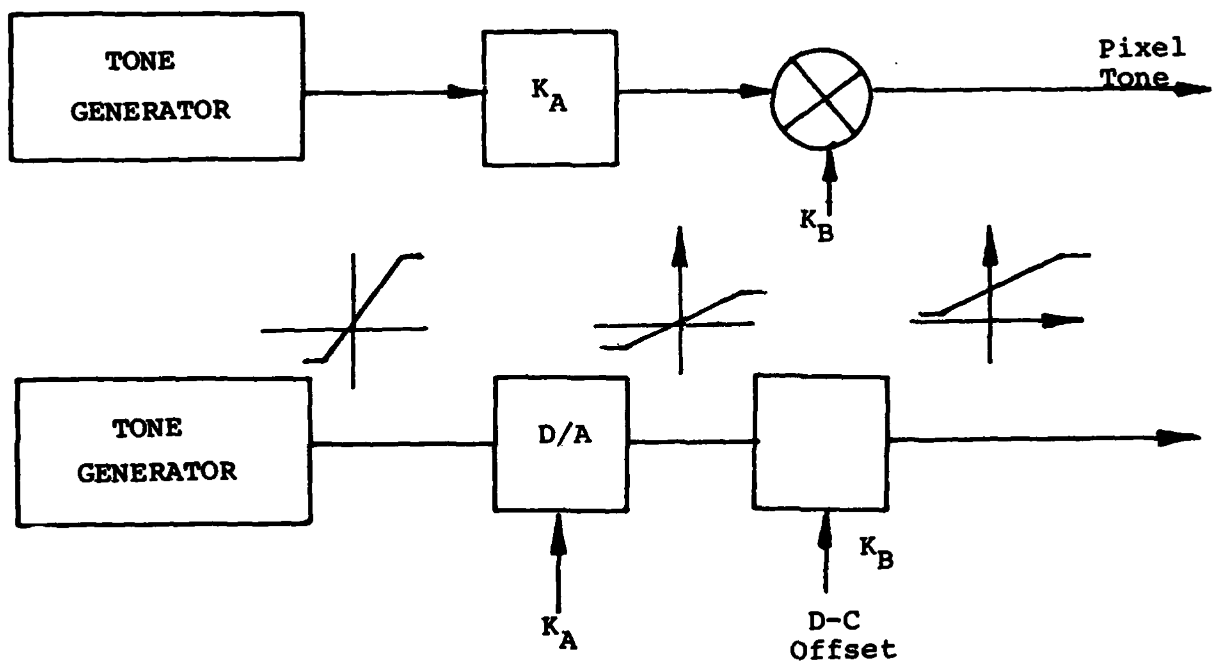


Figure 9. Block Diagram of Linear Estimator

attenuating or scaling the dynamic range of the PRN values (i.e., 255 shades for eighth-degree polynomial), and K_B represents a D-C offset of the resulting pixel tone.

3.2 PIXEL TONE ACCURACY

An analysis was developed to investigate the accuracy of tonal estimates of the simulator. Based on the analysis, a software subsystem was implemented which produces numerical results of the estimations used in the simulator. The software package computes both the exact tone and tone generated by the simulator algorithm for a group of 40 pixels located in the center of a TV line. These two tones are compared and statistics compiled of the exact versus simulated tones for both static and dynamic scenarios.

In addition, specific analysis software has been implemented to pinpoint the exact degree of effects of algorithms.

3.2.1 ANALYSIS DESCRIPTION

The software computes the actual tones for a group of 40 pixels located in the center of a TV line. These actual values are compared to those tones created by the surface texturing simulator. Statistics are kept on the difference between the actual and estimated tones of both static and dynamic scenarios for this 40-pixel window.

The analysis allowed the following conditions to be run:

1. Terrain contrast is a program variable to show the effects of higher contrast scenes.

2. The analysis generates results for up to 30 consecutive TV lines, each 40 pixels wide, to compare line-to-line variations for both static and dynamic scenarios.

3. For the dynamic scenarios, the program collects the results of consecutive frames for the same line in a single report.

4. All of the 8-bit polynomials used in the simulation could be run to develop the statistics for the different PRN generators.

Specific analysis are also performed by the subsystem. These analyses include pixel projection mapping statistical data of each of the eight PRN sequences, and conversion of linear estimator coefficients to integer values.

3.2.2 ANALYTICAL RESULTS

All of the illustrations and reports shown has been generated using the same parameters. Aircraft altitude is 2400 feet; terrain is 6 miles down range; aircraft velocity is 200 knots; the PRN sequence used is the eighth-degree polynomial $x^8 + x^5 + x^3 + x + 1$, terrain contrast is set to 32 shades, and the TV line arbitrarially selected for reporting statistics is line 364.

The analysis included investigations of the following topics:

1. Integer Estimator Coefficients
2. Static Pixel Statistics
   (a) Theoretical Pixel Tone
   (b) Static Scene Statistics, Pixel by Pixel
3. Dynamic Pixel Statistics
4. Linear Estimator Local Accuracies

3.2.2.1 Integer Estimator Coefficients

Within the surface texturing simulator, the coefficients of the linear estimator are multiplied by 2^7 or 128, then converted to integer values. After application of the linear estimator to establish a tone for a particular pixel, the tone is rescaled by multiplying it by 2^{-7}. This is done to increase the speed of the simulator.

Comparison tests were run, correlating the estimated tones of tonal groups using both floating point coefficients and scaled integer coefficients. The differences between the two runs were negligible.

3.2.2.2 Pixel Statistics

One of the capabilities of the software analysis package is to compute the actual tonal value of a given set of pixels and compare this tone to that which is computed by

the surface texturing simulator. The actual value is computed using a tone x area algorithm and includes all whole and any fractional parts of MTA elements which map into a pixel. This value is compared to the value which is computed using the surface texturing simulator. Comparison is done on a window which is 40 pixels wide located in the center of a TV line. The TV line is selectable as an input parameter. The program will compute these statistics for up to 30 consecutive TV lines.

Table 2 presents the results of this analysis phase. Columns of the table have the following meanings:

Pixel: The specific pixel of the 40 the row of data represents (pixel 20 is in the center of the screen).

256 Actual: The true, unscaled tonal value of the pixel element.

32 Actual: The true tonal value of the pixel scaled to 32 shades.

256 Est: The unscaled estimated tonal value of the pixel as computed by the sensor simulator.

32 Est: The estimated tonal value scaled to 32 shades.

256 Del: The difference between the actual and estimated value unscaled; 256 Actual - 256 Est.

32 Del: The difference between the actual and estimated value scaled to 32 shades; 32 Actual - 32 Est.

256 Percent: (256 Del/256) * 100 (Percent error of 256 shades)

32 Percent: Percent error in 32 shades (32 Del/256) * 100

The table in general represents a slightly worse than average case.

The columns labeled 32 Del and 32 Percent are the most meaningful as they represent how typical scenarios appear in the simulator. The least accurate pixel on the page is pixel 39 which is 1.1% or three shades off; this occurred in less than 4 percent of the pixels that were examined.

TABLE 2. SIMULATION ESTIMATION STATISTICS

LINE NUMBER 364

PIXEL	256 ACTUAL	32 ACTUAL	256 EST	32 EST	256 DEL	32 DEL	256 PERCENT	32 PERCENT
1	148	103	136	102	12	1	4.69	0.39
2	152	104	141	102	11	2	4.30	0.78
3	150	103	144	103	6	0	2.34	0.00
4	149	103	152	104	-3	-1	-1.17	-0.39
5	144	103	149	103	-5	0	-1.95	0.00
6	144	103	148	103	-4	0	-1.56	0.00
7	138	102	147	103	-9	-1	-3.52	-0.39
8	144	103	135	101	9	2	3.52	0.78
9	139	102	132	101	7	1	2.73	0.39
10	135	101	133	101	2	0	0.78	0.00
11	136	102	133	101	3	1	1.17	0.39
12	129	101	120	100	9	1	3.52	0.39
13	125	100	113	99	12	1	4.69	0.39
14	127	100	119	99	8	1	3.13	0.39
15	126	100	119	99	7	1	2.73	0.39
16	121	100	115	99	6	1	2.34	0.39
17	113	99	118	99	-5	0	-1.95	0.00
18	116	99	125	100	-9	-1	-3.52	-0.39
19	117	99	132	101	-15	-2	-5.86	-0.78
20	109	98	127	100	-18	-2	-7.03	-0.78
21	107	98	123	100	-16	-2	-6.25	-0.78
22	107	98	111	98	-4	0	-1.56	0.00
23	113	99	115	99	-2	0	-0.78	0.00
24	108	98	112	99	-4	-1	-1.56	-0.39
25	98	97	107	98	-9	-1	-3.52	-0.39
26	92	96	111	98	-19	-2	-7.42	-0.78
27	100	97	108	98	-8	-1	-3.13	-0.39
28	107	98	118	99	-11	-1	-4.30	-0.39
29	111	98	123	100	-12	-2	-4.69	-0.78
30	108	98	120	100	-12	-2	-4.69	-0.78
31	116	99	115	99	1	0	0.39	0.00
32	117	99	115	99	2	0	0.78	0.00
33	120	100	123	100	-3	0	-1.17	0.00
34	129	101	124	100	5	1	1.95	0.39
35	139	102	120	100	19	2	7.42	0.78
36	140	102	124	100	16	2	6.25	0.78
37	135	101	132	101	3	0	1.17	0.00
38	137	102	129	101	8	1	3.13	0.39
39	141	102	116	99	25	3	9.77	1.17
40	141	102	121	100	20	2	7.81	0.78

3.2.2.3 Dynamic Pixel Statistics

Table 3 illustrates a dynamic scenario which compares actual to estimated tones of the 40-pixel window. The numbers are hexadecimal representations of the tonal values. Each row pair represents one frame snapshot of the 40-pixel window for the true pixel tones (top row) and the estimated tones (bottom row). The next row pair shows the tonal value for the next successive frame. The simulation information is the same for this set of numbers as those for Table 2. The numbers show that the data base holds consistent over a dynamic sequence.

The dynamic scenario shows the same statistics as the static pixel statistics. There is a maximum of ±2 shade variations between actual or exact tone and simulated tone.

3.3 SIMULATION RESULTS

After the validity of program operations were verified, algorithms were optimized for improved execution time. A timing model was developed for the simulator. It revealed that subroutine linkage was an impedance which could easily be altered. More importantly, it showed that floating point mathematics instructions in several of the averaging algorithms were consuming the most time; thus, floating point mathematics functions were replaced by integer mathematics functions.

These two enhancements combined to give a savings of more than 50 percent so that typical run times were in the order of 30 to 40 seconds for producing a TV frame.

A six-second scenario was produced which showed the terrain moving smoothly and without scintillation. Of note was the fact that the simulation produced a different tone per pixel on each raster line; this pointed out the need for an extremely high quality monitor that was properly calibrated as to TV raster and interleaving effects.

For speed in execution, table lookup techniques proved to be the best method for defining the basic tones of the PRN structure and starting point of the terrain structure. This technique also allowed the study of subterrain structures.

TABLE 3. DYNAMIC ACTUAL/ESTIMATED COMPARISON

67	68	67	67	67	67	66	67	66	65	66	65	64	64	64	64	63	63	63	62	62	62	63	62	61	60	61	62	62	62	63	63	64	65	66	66	65	66	66	66
66	66	67	66	67	67	67	65	65	65	65	64	63	63	63	63	63	64	65	64	64	62	63	63	62	62	62	63	64	64	63	63	64	64	64	64	65	65	63	64
67	67	66	67	67	67	66	66	66	65	65	65	64	64	64	64	63	63	63	63	62	62	62	63	62	61	60	61	62	62	62	63	63	64	65	66	66	65	66	66
65	65	66	64	65	65	64	66	66	67	67	67	66	64	64	64	63	62	62	62	62	63	64	64	63	63	63	62	63	62	63	63	64	64	66	65	65	66	65	65
67	67	66	67	67	67	66	66	66	66	66	66	65	64	65	64	64	63	63	63	62	62	62	62	62	61	60	61	62	63	62	63	63	64	64	66	66	65	65	66
66	66	67	66	67	67	67	65	65	65	65	65	63	63	63	63	64	65	65	65	64	62	63	63	63	63	63	62	63	64	62	63	64	64	64	64	65	65	64	63
67	67	67	68	67	67	67	66	66	66	66	66	66	64	65	65	64	63	63	63	63	62	62	62	62	62	61	60	62	63	62	62	63	63	64	65	66	66	65	65
66	67	65	65	65	65	65	65	64	64	64	64	66	66	66	66	66	64	63	63	63	63	62	63	62	62	63	63	63	64	64	63	62	63	63	65	65	65	66	66
67	67	67	67	67	67	67	67	66	66	66	66	66	65	64	65	65	64	63	63	63	63	62	62	62	62	61	60	61	62	63	62	63	63	63	64	65	66	66	65
66	66	66	65	66	65	65	67	66	67	67	67	66	65	64	64	64	62	62	62	62	64	64	64	64	64	63	61	61	62	64	63	63	64	64	64	64	65	65	65
67	67	67	67	66	67	67	66	66	66	66	66	66	66	64	64	65	64	64	63	63	63	62	62	62	62	62	61	60	61	62	63	62	63	63	64	65	66	66	65
66	66	67	67	67	67	67	65	64	65	64	64	64	64	64	63	64	66	65	65	65	63	63	63	63	63	62	62	62	62	62	63	64	64	64	65	64	64	64	64
66	67	67	67	67	67	67	67	67	66	66	66	65	66	65	64	64	64	64	63	63	63	63	62	62	62	62	61	60	60	62	62	62	63	63	63	64	66	66	66
67	67	65	66	65	65	65	65	65	64	64	64	65	67	67	66	66	65	64	64	63	63	62	62	62	63	63	63	64	63	63	63	62	63	62	64	64	65	66	66
66	67	67	67	67	66	67	67	67	67	66	67	66	65	66	65	64	64	64	64	63	63	63	62	62	62	63	62	61	60	61	62	62	62	63	63	64	65	66	66
67	67	65	66	66	66	66	66	67	67	67	67	66	65	65	65	63	63	63	63	62	63	64	64	64	64	63	63	62	61	62	63	64	64	64	63	64	64	64	64
66	66	67	67	67	67	66	67	67	67	66	66	66	66	65	65	64	64	64	64	63	63	63	63	62	62	62	63	62	61	60	61	62	62	62	63	63	64	65	66
66	67	66	66	65	65	65	64	65	65	64	66	65	67	67	67	66	65	64	64	64	63	62	62	62	62	64	64	63	64	63	62	63	62	62	63	64	64	66	66
66	66	66	67	67	67	66	67	67	67	66	66	66	66	66	66	65	64	65	64	64	63	63	63	62	62	62	62	62	61	60	61	62	63	62	63	63	64	64	66
67	67	66	66	66	66	66	66	67	67	67	67	65	65	65	65	63	63	63	63	63	63	64	65	64	63	62	63	63	62	63	62	63	64	63	63	64	64	64	64
66	66	66	67	67	67	67	66	67	67	67	66	66	66	66	66	66	64	65	65	64	63	63	63	63	62	62	62	62	62	61	60	62	63	62	62	63	63	64	65
66	66	66	67	66	67	66	65	65	65	65	64	65	64	64	64	66	66	66	66	66	64	64	63	63	63	63	63	62	62	63	63	63	64	64	63	62	63	63	64
67	66	66	66	67	67	67	67	67	67	67	67	66	66	66	66	66	65	64	65	65	64	63	63	63	63	62	62	62	62	62	60	61	62	63	62	63	63	63	64
65	66	66	67	66	66	65	65	65	65	65	66	66	67	67	67	65	65	64	64	64	64	62	62	62	63	63	64	64	64	63	62	61	62	64	63	63	64	64	64
66	67	66	66	67	67	67	67	66	67	67	66	67	66	66	66	66	66	64	64	65	65	64	63	63	63	62	62	62	62	62	61	60	61	62	63	62	63	63	64
66	66	66	67	66	66	66	67	67	67	67	67	64	65	64	65	64	64	64	63	63	64	65	65	65	63	62	63	63	63	62	63	62	62	63	63	64	64	64	64

3.3.1 TEXTURING

Texture data base sizes were selected such that the terrain was typically limited to 1.5 nautical miles in length (dependent on MTA sizes). There was no limitation on terrain width. The program could be altered to expand to larger terrain lengths up to the limits of available user memory on the operating computer. Typical terrains investigated were between 600 feet to 2 nautical miles wide and 1.5 miles in length.

3.3.1.1 Texture Types

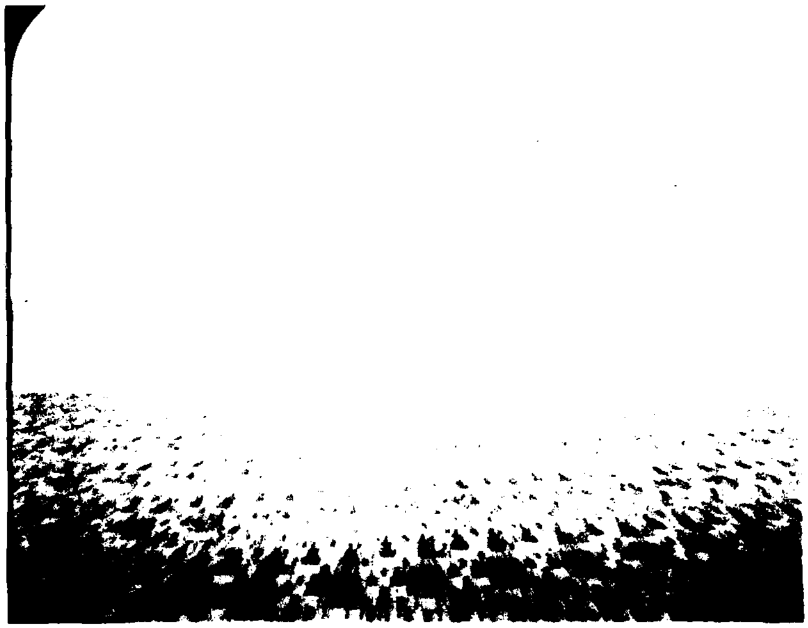
Figure 10 is a set of still photographs that show different texture patterns using the same PRN polynomial. The photos were taken with extremely high contrast on the display so that they could be reproduced; the actual visual display can be set with less contrast, which is more pleasing to the viewer. Some effect of contrast of the texture will be described later in this section.

Table 4 defines the scenario-simulation parameters for Figure 10. This figure demonstrates the versatility of PRN codes as a source of texturing data. All of the pictures shown in the figures were created by the same eighth-degree polynomial generator. The variation that is shown depicts various forms of data base structure and subterrain structure. Figure 10 (b-c) show mosaic-type structures while 10d depicts a more continuously modulating structure. It is believed that greater variations in texture type can be obtained by usage of different and larger polynomial generators, different subterrain structures and various shading schemes. A detailed investigation of the various mosaic and continuous terrain types was beyond the scope of this study but is recommended for future study.

Figure 10a shows the basic texture created by an 8-bit PRN polynomial generator with the stepped algorithm subterrain structure. Here the subterrains are clearly visible. Review of the photograph will show the basic pseudo-random pattern. Several bands are wide, while others are very narrow. Some bands change gradually from dark to light (and vice versa) while others change abruptly. In this particular photograph, the 8-bit generator does not repeat itself in the horizontal direction but does repeat itself several times in the vertical direction.

Review of the photographs on the actual ACTES display shows the repeating pattern in the terrain background but with gradual loss of foreground detail. The photographs in

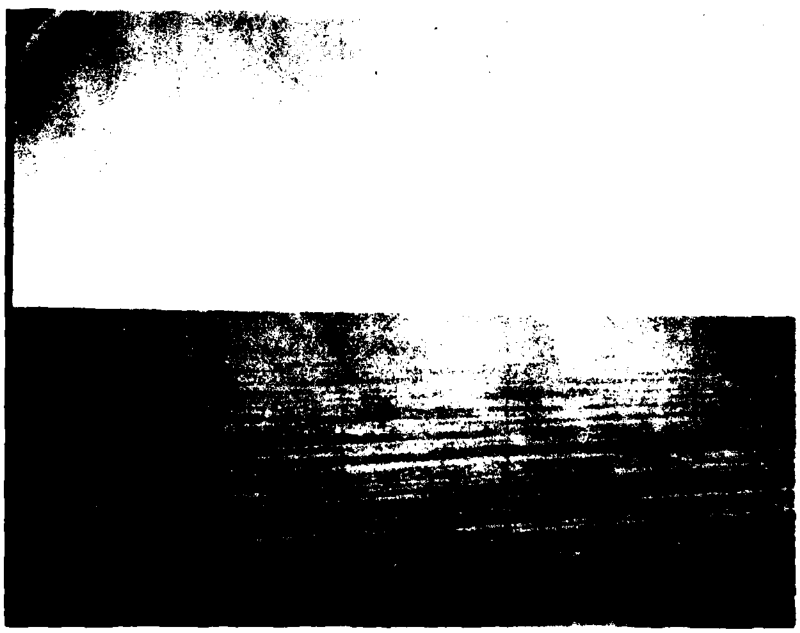
10a.



10b.

Figure 10. Texture Patterns

10c.



10d.

Figure 10 (continued)

TABLE 4. SCENARIO-SIMULATION PARAMETERS

Figure No.	Data Base			Code/Display	Aircraft Position
	Terrain Size L x W (nm)	MTA Ft. x Ft.	Subterrain Structure L1 x L2 x L3 *	Tonal Levels Attenuation	Alt * Range Ft * nm
10a	2 x 2	10 x 10	50 x 50 x 1	32	200 x 2
10b	1.5 x .1	6 x 6	50 x 25 x 0	32	300 x 2
10c	1.6 x 3	6 x 6	25 x 25 x 0	32	300 x 2
10d	1.6 x 3	6 x 6	25 x 25 x 0	32	1000 x 3
11a				32	
11b				16	
11c	1.5 x .1	10 x 10	50 x 50 x 1	8	300 x 2
11d				4	
12a	1.5 x .1	6 x 6	50 x 50 x 1	32	1000 x 5
12b	1.5 x .1	6 x 6	50 x 50 x 1	32	1000 x 3
12c	1.5 x .1	6 x 6	50 x 50 x 1	32	300 x 2
12d	1.5 x .025	10 x 10	20 x 64 x 1	32(4)	500 x 2

* L1 = No MTA in Subterrain Length
L2 = Slip rate between Subterrain Starting States
L3 = Slip rate of Starting States within Subterrain

Figure 10 also show the results of the combinatorial algorithms and data base quantization. Figure 10b shows the effects of data base quantization. At the bottom of the terrain cne MTA-PRN tone covers more than one pixel. At the top of the terrain, many MTA-PRN tones are averaged to form the pixel tone. Depending on the mission flyby geometry, the minimum texture area (MTA) should be selected such that the simulation is always averaging many PRN tones per pixel to eliminate a stylized display, thereby providing more display realism. Intra-line pixel averaging could also be implemented to smooth this effect if it proves to be a desirable effect for terrain cueing.

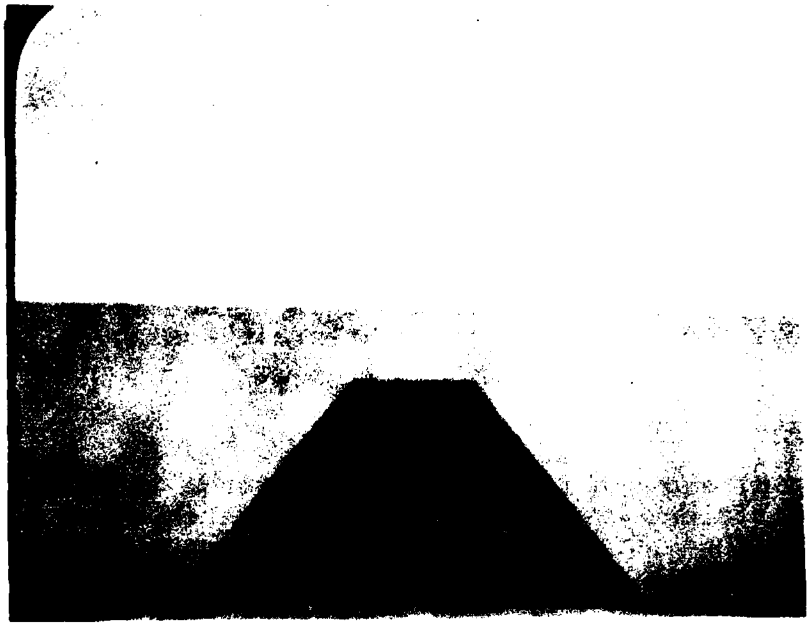
3.3.1.2 Texture Contrast

Figure 11 (a-d) shows a terrain at close range (300 ft., 2 nm) with different gray scale levels or attenuation of the tonal dynamic range within the terrain structure. Table 4 describes the simulation parameters. The original 255 shades generated by the PRN generator are scaled down to 32, 16, 8 and 4 shades, respectively, for the four photographs. It is easily seen how the texture becomes smoother or undulating as the number of shades decreases. Increasing the number of shades results in a more detailed terrain image with sharp contrast levels. This permits greater flexibility in terrain modeling and possibly in perception and distance cueing. The effect is that as the number of shades increases, the terrain appears to get closer to the viewer as it is more detailed, regardless of the fact that the images do not increase in size.

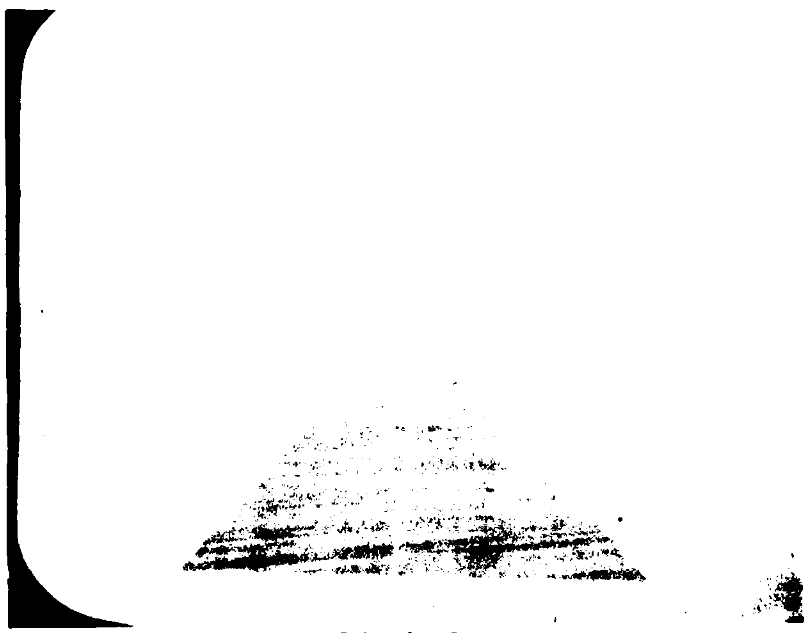
3.3.1.3 Texture Detail

One of the most important concepts shown by this study is the effect of the PRN averaging techniques on perception and texture cueing. The viewing of a recognizable "texture" pattern which increases in detail as the range from viewer to terrain decreases is a fundamental to flight training and target discrimination applications.

Figure 12 (a through d) shows the effect on textural perception as the terrain approaches the aircraft sensor. Figure 12a shows the view of the terrain from an altitude of 1000 feet and range of 5 nautical miles. In Figure 12b, the terrain is 3 nautical miles in front of the aircraft at the same altitude. Note that in the texture area of Figure 12b, the texture pattern can now be observed. In Figure 12c, the aircraft is at an altitude of 300 feet and

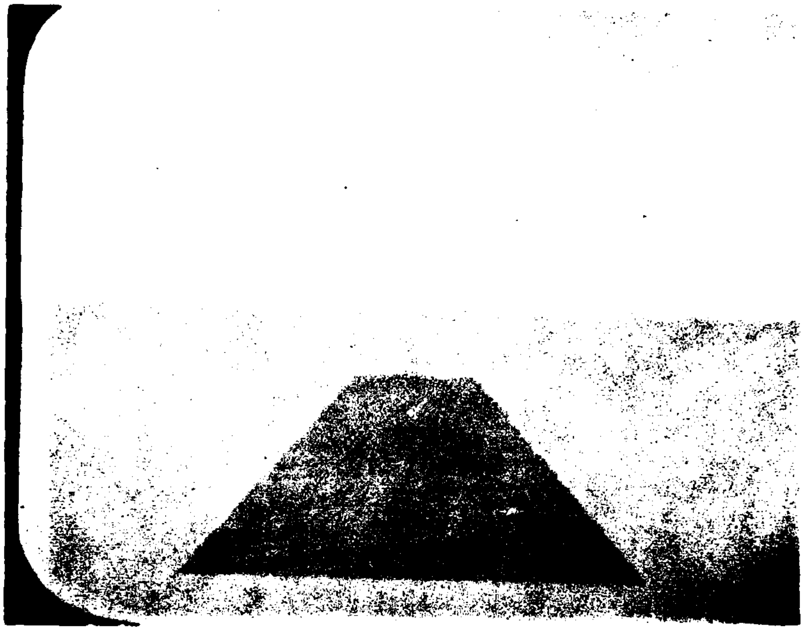


11a. 32 Shades

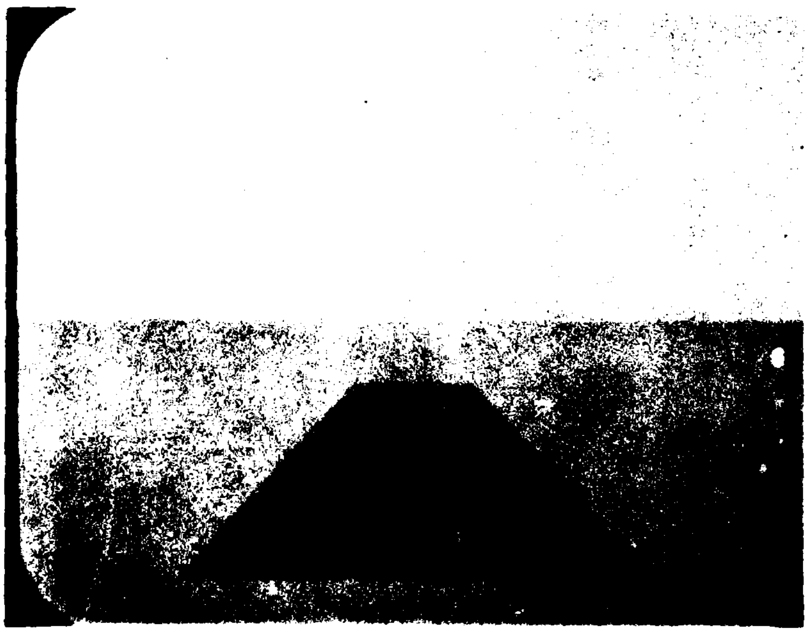


11b. 16 Shades

Figure 11



11c. 8 Shades



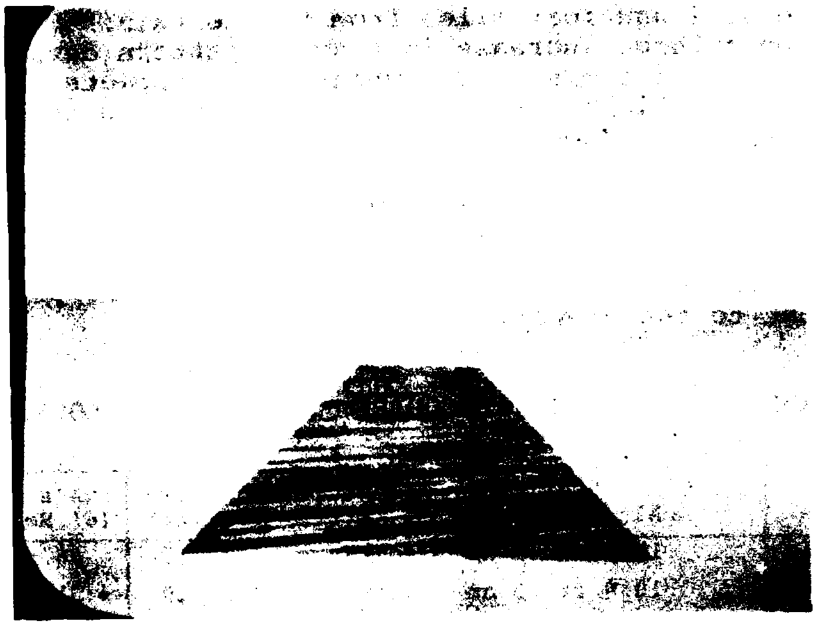
11d. 4 Shades

Figure 11 (continued)

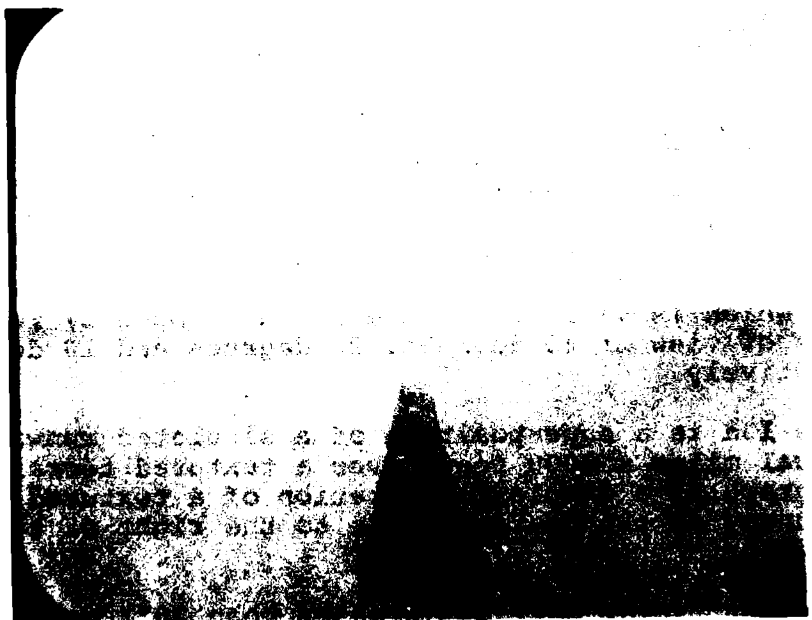
12a. Range = 5 miles
Altitude = 1000 feet

12b. Range = 3 miles
Altitude = 1000 feet

Figure 12



12c. Range = 2 miles
Altitude = 300 feet



12d. Long Strip

Figure 12 (continued)

a range of 2 nautical miles from the terrain. From this position a large increase in texture pattern detail can be seen. Table 5 summarizes combinatorial aspects of the simulation, which is required in each case to develop the video representation.

The relationship of texture detail to distance is also illustrated in Figure 12d. In this long terrain, the texture pattern is visible close to the viewer but is lost in distance with only the general texture pattern apparent to the viewer.

TABLE 5. AVERAGING REQUIREMENTS OF PRN ALGORITHM

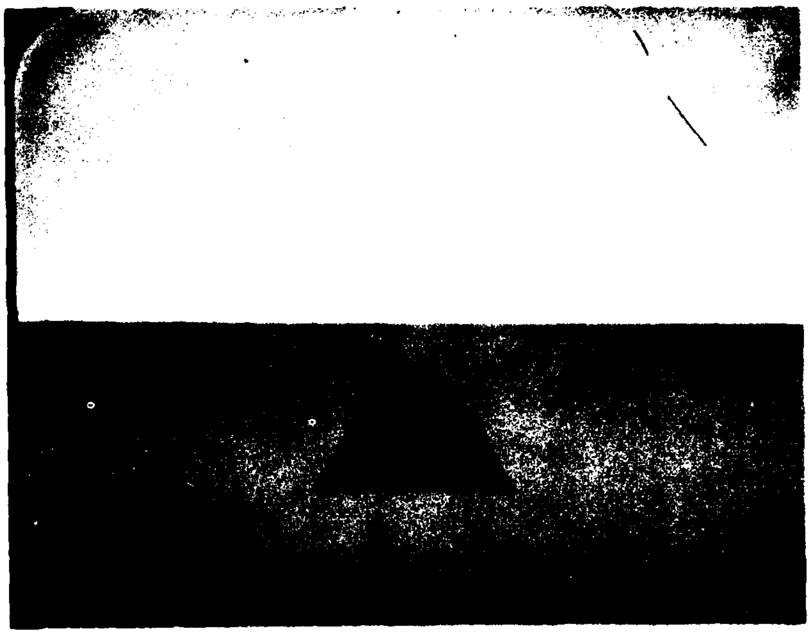
Figure	Altitude/Range	Texture Area/Pixel ΔX(ft)	ΔY(ft)	MTA's Averaged (6' Sq. MTA's)
12a	1000 ft 5 nm	515	17	86 x 3 = 258
12b	1000 ft 3 nm	184	10	31 x 2 = 62
12c	300 ft 2 nm			
	Top (Background)	275	7	46 x 1 = 46
	Bottom (Foreground)	16.9	1.7	3 x 1 = 3

3.3.2 FIELD OF VIEW CONSIDERATIONS

Figure 13 (a-c) shows the effect of a change in the field of view. The terrain data base in all cases is exactly the same. It is positioned the same distance from the viewpoint in all instances. Field of view is the only parameter which is varied. Figure 13 (a through c) represents fields of view of 45 degrees, 20 degrees and 10 degrees, respectively.

Figure 13d is a superposition of a simulated runway (1.5 nautical miles x 3000 feet) over a textured terrain. The photograph is a simple illustration of a textured scene. The runway is .25 nautical mile to the right of the viewpoint.

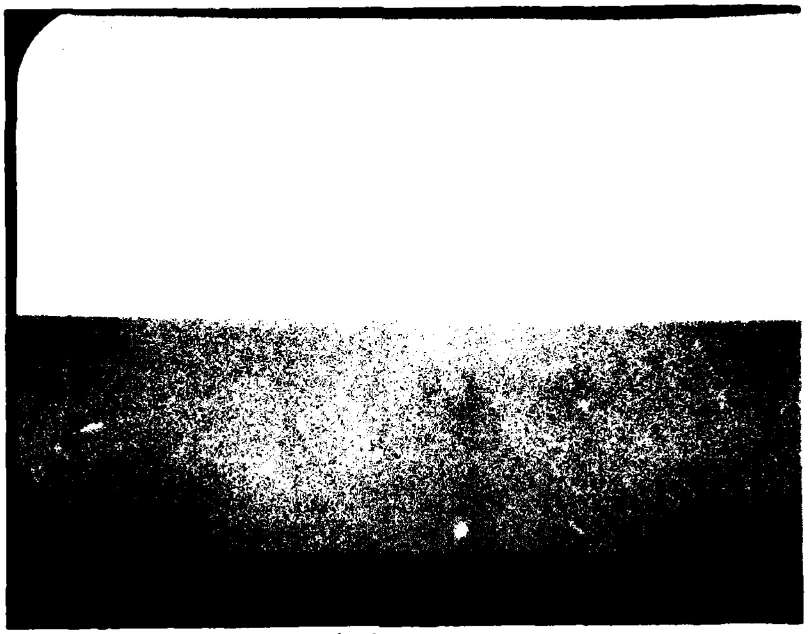
The larger fields of view (greater than 20°) were run to demonstrate the applicability of the technique to provide texturing to visual CIG displays. Because the simulation algorithms provided the same types of presentations as for narrow fields of view, these techniques can probably be used for visual presentations in wide FOV simulation applications.



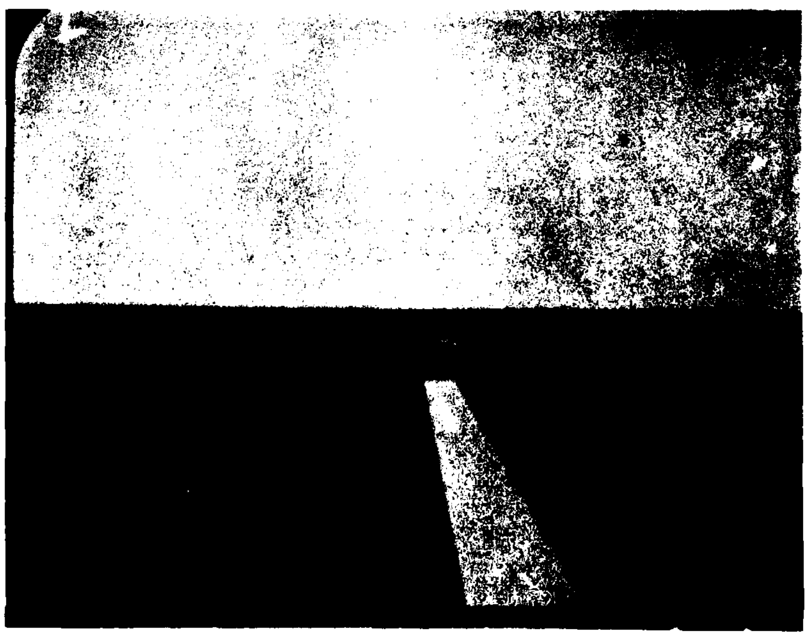
13a. Field of View 45°

13b. Field of View 20°

Figure 13



13c. Field of View 10°



13d. Runway in a Field

Figure 13 (continued)

3.3.3 THREE-DIMENSIONAL ORIENTATION

The simulation was enhanced to provide six degrees of freedom of the viewpoint with respect to the terrain. The terrain edges in the object space rotated according to the angular attitude of the field of view. The texturing patterns were also rotated or skewed in a similar manner.

3.4 ACTES FACILITY OPERATIONS

The simulation was delivered, implemented and run on the Simulation and Training Advanced Research System (STARS) ACTES Xerox Sigma 5 computer system located at Wright-Patterson AFB. The Sigma 5 computer ran the actual simulation and stored the video data on magnetic tape. The offline data storage retrieval and display capabilities of the ACTES facility were used to generate and display the flyby scenarios.

The generation and storage of video data from the Sigma 5 onto magnetic tape have limitations. Due to the format required for the ACTES offline capabilities, approximately 30 TV frames or one second of display would fit on a tape. The combined lengthiness of the computer operations (tape mounting, rewind, etc.) impeded the generation of long simulations. It is recommended that the direct Sigma 5 - ACTES computer interface be investigated to speed up the mechanical process of generating scenarios.

Programs were initially delivered on cards and stored on disk. Online terminals are provided for source file updating, program compilations and simulation data input/output and control in an interactive mode. The high speed terminals enabled easy access to source files and made the interactive simulation possible and simple to operate.

4.0 CONCLUSIONS AND RECOMMENDATIONS

4.1 CONCLUSIONS

This study showed the versatility and capabilities of using PRN coding techniques for generating texture patterns in CIG-type simulations. Variation in texture detail was demonstrated and showed a smooth transition from foreground detail to background general patterns. A number of terrain types, including mosaic-type areas and continuous undulating areas, were developed showing the texturing capability of the techniques. Scenes for different fields of view were analyzed and indicated that the technique can be extended to include visual-type displays as well as sensor-type displays.

The techniques implemented in this non-real-time software simulation are adaptable to real-time hardware.

During playback on the TV monitor, areas of scintillation-type variations were noticed in both the terrain texturing and the monotone gray areas. Analytical results of a line of TV pixels show there is a maximum variation of ±2 gray shades within the textured terrain between exact and simulated tones. It is believed that this variation would be difficult to detect within the texture. Because the monotone gray area had similar variations, no conclusions could be made regarding the source of the variations.

Due to the computation time required to create a scenario, long scenarios were not created in the Sigma 5. This would have assisted in the determination of the source of these variations. It is recommended that a partial or total hardware implementation of the tonal calculation be made, thus reducing scene computation time and enabling closer analysis of the video display effects.

4.2 SYSTEM ENHANCEMENTS AND RECOMMENDATIONS

One recommended change is a hardware adaptation to the ACTES system to perform the tonal calculations in real time. The simulation software can be made to function within the ACTES.

An alternate recommendation is a partial or total hardware implementation in a high-speed data input/output mode of the tonal generation presently being performed in software. This implementation will increase the accuracy and speed of the tonal calculations, thereby providing the capability to produce long scenarios. Long scenarios will enable a

visual analysis of the variations presently being viewed on the TV monitor. This hardware could become a model for real-time hardware specifications.

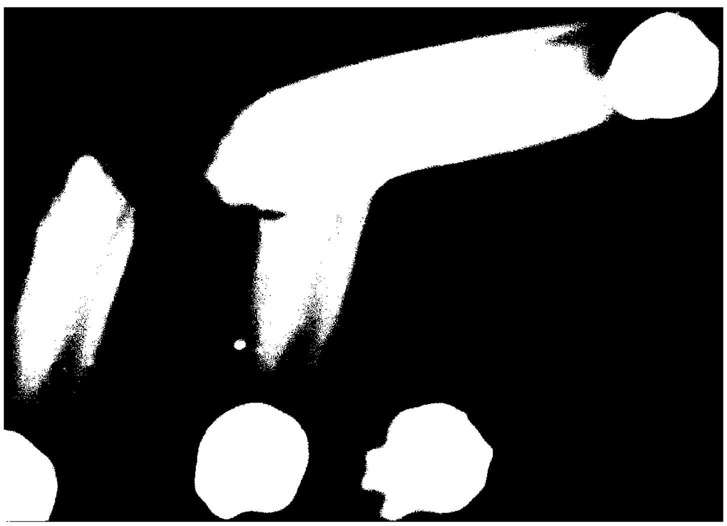
A recommended enhancement to the system is the ability to provide three-dimensional texturing in the object space. Completion of this enhancement would then allow the use of the cyclic coding techniques to be considered for inclusion in existing non-real-time and future real-time simulation systems.

Further investigation of cyclic code generators should attempt to correlate texture patterns to realistic terrain environments, such as forests, fields, grass, etc. This effort would include the use of larger code generators, different and potentially non-maximal-length polynomials, subterrain structures, and superimposed multiple code sequences.

ND

ATE

LMED



AD-A093 734 APPLIED DIGITAL COMMUNICATIONS MOORESTOWN NJ F/G 5/8
FLIGHT TRAINING SIMULATOR: SURFACE TEXTURING VIA PSEUDO RANDOM --ETC(U)
DEC 80 J F DESPAUTZ, M B BENDER, V M MCNAMARA F33615-77-C-0068
UNCLASSIFIED AFHRL-TR-80-13 NL

2 OF 2

AD A
093734

SUPPLEMENTARY INFORMATION

END
DATE FILMED
6-81
DTIC

AD
09

SUPPLEMENTARY

INFORMATION

DEPARTMENT OF THE AIR FORCE
AIR FORCE HUMAN RESOURCES LABORATORY (AFSC)
BROOKS AIR FORCE BASE, TEXAS 78235

Errata

REPLY TO ATTN OF: TSR

16 JAN 1981

SUBJECT: Removal of Export Control Statement

TO: Defense Technical Information Center
Attn: DTIC/DDA (Mrs Crumbacker)
Cameron Station
Alexandria VA 22314

AD-A093734

1. Please remove the Export Control Statement which erroneously appears on the Notice Page of the reports listed [redacted]. This statement is intended for application to Statement B reports only.

2. Please direct any questions to AFHRL/TSR, AUTOVON 240-3877.

FOR THE COMMANDER

Wendell L Anderson

WENDELL L. ANDERSON, Lt Col, USAF
Chief, Technical Services Division

1 Atch
List of Reports

Cy to: AFHRL/TSE

EN

DAT
FILME

6 - 8

DTI